

Rec'd 12 MAY 2005

10/534750

PCT/JP 03/15274

日本国特許庁
JAPAN PATENT OFFICE

28.11.03

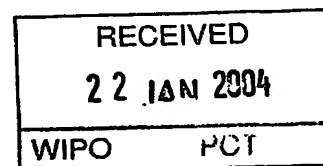
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年11月29日

出願番号
Application Number: 特願2002-348726
[ST. 10/C]: [JP 2002-348726]

出願人
Applicant(s): キヤノン株式会社

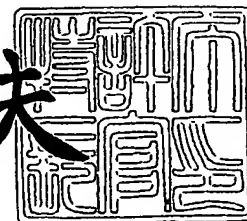


PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2004年 1月 7日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3109045

【書類名】 特許願

【整理番号】 226551

【提出日】 平成14年11月29日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00

【発明の名称】 情報処理方法及び装置

【請求項の数】 34

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 藤木 真和

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 大島 登志一

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100076428

 【弁理士】

 【氏名又は名称】 大塚 康德

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100112508

 【弁理士】

 【氏名又は名称】 高柳 司郎

 【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100116894

【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理方法及び装置

【特許請求の範囲】

【請求項 1】 情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法であって、

前記共有データに対する操作要求が発生した場合に、該操作要求を表す要求情報を前記情報伝達媒体上に出力する出力工程と、

前記出力工程によって出力された前記要求情報に対する応答情報を前記情報伝達媒体より受信する受信工程と、

前記受信工程で受信された応答情報に従って前記共有データを操作する操作実行工程とを備えることを特徴とする情報処理方法。

【請求項 2】 前記操作実行工程は、前記応答情報を受信する前に前記操作要求の発生から所定時間が経過した場合には、該応答情報の受信を待たずに前記操作要求に基づく前記共有データの操作を実行することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 3】 前記操作要求に応じて操作キューにキューアイテムを登録し、前記操作実行工程によって前記共有データに対する操作が実行された場合に対応するキューアイテムを処理済とするキュー制御工程を更に備え、

前記操作実行工程は、前記操作キューにおいて前記応答情報が表す操作要求に対応するアイテムが処理済となっていない場合に、該応答情報に対応する操作を前記共有データに対して実行することを特徴とする請求項 2 に記載の情報処理方法。

【請求項 4】 前記操作実行工程は、

前記受信工程によって前記応答情報が受信されるのを待って前記共有データの操作を実行する第 1 モードと、

前記操作要求の発生からの所定時間の経過と、前記受信工程による前記応答情報の受信の何れかの早い方のタイミングで前記共有データの操作を実行する第 2

モードを含む複数の変更モードのいずれかで動作することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 5】 前記共有データは複数のアイテムで構成され、該複数のアイテムの各々は採用すべき変更モードを指定する指定情報を含むことを特徴とする請求項 4 に記載の情報処理方法。

【請求項 6】 前記複数のアイテム毎に変更モードを切り換える切り換え工程を更に備えることを特徴とする請求項 5 に記載の情報処理方法。

【請求項 7】 前記切り換え工程は、所望のアイテムに対応するオブジェクト表示を選択して所望の変更モードを指定するためのユーザインターフェースを提供することを特徴とする請求項 6 に記載の情報処理方法。

【請求項 8】 前記切り換え工程による変更モードの切り換え結果が、前記複数のクライアントの共有データに反映されることを特徴とする請求項 6 に記載の情報処理方法。

【請求項 9】 前記切り換え工程による変更モードの切り換え結果が、当該クライアントの共有データのみ反映されることを特徴とする請求項 6 に記載の情報処理方法。

【請求項 10】 前記所定時間を設定する設定工程を更に備えることを特徴とする請求項 4 に記載の情報処理方法。

【請求項 11】 前記操作実行工程は、前記応答情報を受信する前に、システムが定める更新過程の開始時刻に達した場合には、該応答情報の受信を待たずに前記操作要求に基づく前記共有データの操作を実行することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 12】 前記切り換え工程は、アイテムに対応するオブジェクトに対する操作内容に応じて変更モードを設定することを特徴とする請求項 6 に記載の情報処理方法。

【請求項 13】 情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法であって、

複数のクライアントプロセスとの接続を確立する確立工程と、
前記複数のクライアントプロセスより共有データの変更に関するイベントを受信する受信工程と、

前記受信工程で受信したイベントを、前記複数のクライアントプロセスに対して発行する発行工程とを備えることを特徴とする情報処理方法。

【請求項 14】 前記受信工程で受信するイベントは共有データの変更手順を示す変更モード情報を含み、

前記発行工程は前記変更モード情報に基づいて前記イベントの配信先を制御することを特徴とする請求項 13 に記載の情報処理方法。

【請求項 15】 前記発行工程は、前記受信工程で受信したイベントの変更モード情報に基づいて当該イベントの発行元において共有データの処理が既に完了しているか判定し、完了している判定された場合には、当該イベントをその発行元のクライアントへは発行しないことを特徴とする請求項 14 に記載の情報処理方法。

【請求項 16】 前記イベントはタイムアウト時刻を更に含み、

前記判定は、前記変更モード情報がタイムアウト時刻の経過によって操作が実行される変更モードであることを示し、且つ、現在時刻がタイムアウト時刻を過ぎている場合に、当該イベントの発行元において共有データの処理が既に完了していると判定することを特徴とする請求項 15 に記載の情報処理方法。

【請求項 17】 情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理装置であって、

前記共有データに対する操作要求が発生した場合に、該操作要求を表す要求情報を前記情報伝達媒体上に出力する出力手段と、

前記出力手段によって出力された前記要求情報に対する応答情報を前記情報伝達媒体より受信する受信手段と、

前記受信手段で受信された応答情報に従って前記共有データを操作する操作実行手段とを備えることを特徴とする情報処理装置。

【請求項 18】 前記操作実行手段は、前記応答情報を受信する前に前記操作要求の発生から所定時間が経過した場合には、該応答情報の受信を待たずに前記操作要求に基づく前記共有データの操作を実行することを特徴とする請求項 17 に記載の情報処理装置。

【請求項 19】 前記操作要求に応じて操作キューにキューアイテムを登録し、前記操作実行手段によって前記共有データに対する操作が実行された場合に対応するキューアイテムを処理済とするキュー制御手段を更に備え、

前記操作実行手段は、前記操作キューにおいて前記応答情報が表す操作要求に対応するアイテムが処理済となっていない場合に、該応答情報に対応する操作を前記共有データに対して実行することを特徴とする請求項 18 に記載の情報処理装置。

【請求項 20】 前記操作実行手段は、

前記受信手段によって前記応答情報が受信されるのを待って前記共有データの操作を実行する第 1 モードと、

前記操作要求の発生からの所定時間の経過と、前記受信手段による前記応答情報の受信の何れかの早い方のタイミングで前記共有データの操作を実行する第 2 モードを含む複数の変更モードのいずれかで動作することを特徴とする請求項 17 に記載の情報処理装置。

【請求項 21】 前記共有データは複数のアイテムで構成され、該複数のアイテムの各々は採用すべき変更モードを指定する指定情報を含むことを特徴とする請求項 20 に記載の情報処理装置。

【請求項 22】 前記複数のアイテム毎に変更モードを切り換える切り換え手段を更に備えることを特徴とする請求項 21 に記載の情報処理装置。

【請求項 23】 前記切り換え手段は、所望のアイテムに対応するオブジェクト表示を選択して所望の変更モードを指定するためのユーザインターフェースを提供することを特徴とする請求項 22 に記載の情報処理装置。

【請求項 24】 前記切り換え手段による変更モードの切り換え結果が、前記複数のクライアントの共有データに反映されることを特徴とする請求項 22 に記載の情報処理装置。

【請求項 25】 前記切り換え手段による変更モードの切り換え結果が、当該クライアントの共有データのみには反映されることを特徴とする請求項 22 に記載の情報処理装置。

【請求項 26】 前記所定時間を設定する設定手段を更に備えることを特徴とする請求項 20 に記載の情報処理装置。

【請求項 27】 前記操作実行手段は、前記応答情報を受信する前に、システムが定める更新過程の開始時刻に達した場合には、該応答情報の受信を待たずに前記操作要求に基づく前記共有データの操作を実行することを特徴とする請求項 17 に記載の情報処理装置。

【請求項 28】 前記切り換え手段は、アイテムに対応するオブジェクトに対する操作内容に応じて変更モードを設定することを特徴とする請求項 22 に記載の情報処理装置。

【請求項 29】 情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理装置であって、

複数のクライアントプロセスとの接続を確立する確立手段と、

前記複数のクライアントプロセスより共有データの変更に関するイベントを受信する受信手段と、

前記受信手段で受信したイベントを、前記複数のクライアントプロセスに対して発行する発行手段とを備えることを特徴とする情報処理装置。

【請求項 30】 前記受信手段で受信するイベントは共有データの変更手順を示す変更モード情報を含み、

前記発行手段は前記変更モード情報に基づいて前記イベントの配信先を制御することを特徴とする請求項 29 に記載の情報処理装置。

【請求項 31】 前記発行手段は、前記受信手段で受信したイベントの変更モード情報に基づいて当該イベントの発行元において共有データの処理が既に完了しているか判定し、完了している判定された場合には、当該イベントをその発行元のクライアントへは発行しないことを特徴とする請求項 30 に記載の情報

処理装置。

【請求項 3 2】 前記イベントはタイムアウト時刻を更に含み、

前記判定は、前記変更モード情報がタイムアウト時刻の経過によって操作が行われる変更モードであることを示し、且つ、現在時刻がタイムアウト時刻を過ぎている場合に、当該イベントの発行元において共有データの処理が既に完了していると判定することを特徴とする請求項 3 1 に記載の情報処理装置。

【請求項 3 3】 請求項 1 乃至 1 6 のいずれかに記載の情報処理方法コンピュータに実行させるための制御プログラム。

【請求項 3 4】 請求項 1 乃至 1 6 のいずれかに記載の情報処理方法コンピュータに実行させるための制御プログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数のプロセス間でデータを共有する技術に関する。特に 3 次元仮想空間を記述するデータを保持するデータベースをネットワークを介して複数のプロセス間で同期させるのに好適な技術に関する。

【0002】

【従来の技術】

異なるコンピュータ端末間で 3 次元仮想空間を共有する技術は、遠隔会議システム、ネットワーク型対戦ゲーム、協調型デザインシステムなどを実現するために必須である。

【0003】

このような 3 次元仮想空間を共有するシステムでは、仮想空間の動画像をコンピュータグラフィクスにより描画するために、短い時間間隔で多量のデータを参照する。このとき、一つの装置（例えばサーバ）に保持されている仮想空間のデータ全体をネットワークを介して取得することは、現在のネットワークのバンド幅の狭さから現実的ではない。したがって、各々の端末が仮想空間データのコピーを持ち、それを参照することによって CG 画像を生成する方式が採用されている。また、あるコンピュータ端末上で仮想空間に対して何らかの操作、例えば仮

想物体の移動や回転が行われた場合には、その操作に関する情報が他の端末にネットワーク経由で伝送され、各端末の持つデータベースに反映される。この結果、各コンピュータ端末におけるデータベースの同一性が保たれる。

【0004】

従来の仮想空間共有システムにおける上述のデータベースの同一性保持の手順の詳細は次のとおりである。すなわち、あるコンピュータ端末における仮想空間に対する操作が当該端末のローカルのデータベースに直ちに反映されるとともに、当該操作情報が他のコンピュータ端末に送信される。他の端末は、この操作情報を受信し、受信した操作情報に従ってデータを更新する。

【0005】

このような仮想空間共有システムの実現例としては、非特許文献1に記載されたものが挙げられる。

【0006】

【非特許文献1】

Distributed Open Inventor (出展：G.Heshina et.al. : "Distributed Open Inventor : A Practical Approach to Distributed 3D Graphics", in Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST'99), pp.74-81, 1999)

【0007】

【発明が解決しようとする課題】

しかしながら、従来の仮想空間共有システムでは、操作を行なった端末においてその操作内容が当該端末のデータベースに即座に反映され、他の端末はその操作情報を受信してからデータベースの更新が行なわれる。従って、操作を行った端末とそれ以外の端末とでデータベースを更新するタイミングが異なってしまう。特に、ネットワーク上のトラフィックの影響等で、操作情報の受信までに時間がかかると、端末間のデータベースの更新のタイミングのズレは顕著となる。このため、端末間で仮想空間の描画結果に違いが生ずる可能性があるという問題点があった。

【0008】

また、データベースの操作および更新の手順は上記の如き手順に固定されており、データ更新のタイミングを選択あるいは調整することができないという問題もある。

【0009】

本発明は上記の課題に鑑みてなされたものであり、複数のプロセスで共有される共有データに関して、操作を実行するタイミングをほぼ同一とすることを目的とする。

【0010】

【課題を解決するための手段】

上記の課題を解決するための本発明による情報処理方法は、

情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法であって、

前記共有データに対する操作要求が発生した場合に、該操作要求を表す要求情報を前記情報伝達媒体上に出力する出力工程と、

前記出力工程によって出力された前記要求情報に対する応答情報を前記情報伝達媒体より受信する受信工程と、

前記受信工程で受信された応答情報に従って前記共有データを操作する操作実行工程とを備える。

また、上記の課題を解決するための本発明による他の態様の情報処理方法は、

情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法であって、

複数のクライアントプロセスとの接続を確立する確立工程と、

前記複数のクライアントプロセスより共有データの変更に関するイベントを受信する受信工程と、

前記受信工程で受信したイベントを、前記複数のクライアントプロセスに対して発行する発行工程とを備える。

更に、上記の目的を達成するための本発明による情報処理装置は以下の構成を

備える。すなわち、

情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理装置であって、

前記共有データに対する操作要求が発生した場合に、該操作要求を表す要求情報を前記情報伝達媒体上に出力する出力手段と、

前記出力手段によって出力された前記要求情報に対する応答情報を前記情報伝達媒体より受信する受信手段と、

前記受信手段で受信された応答情報に従って前記共有データを操作する操作実行手段とを備える。

更に、上記の目的を達成するための本発明による他の態様の情報処理装置は以下の構成を備える。すなわち、

情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理装置であって、

複数のクライアントプロセスとの接続を確立する確立手段と、

前記複数のクライアントプロセスより共有データの変更に関するイベントを受信する受信手段と、

前記受信手段で受信したイベントを、前記複数のクライアントプロセスに対して発行する発行手段とを備える。

【0011】

【発明の実施の形態】

以下、添付の図面を参照して本発明の好適な実施形態を説明する。

【0012】

〔第1実施形態〕

第1実施形態では、本願発明による情報処理方法を、仮想空間の構造と属性を記述するシーンデータベースを複数の端末で共有する仮想空間共有システムに応用した実現形態について説明する。

【0013】

本実施形態の仮想空間共有システムの全体的な構成を図1に示す。システムには1つのサーバプロセス101（以下、「サーバ」と略す）と、複数のクライアントプロセス103（以下、「クライアント」と略す）があり、ネットワーク102を介してデータを交換する。それぞれのクライアントは共通の仮想空間の構造と属性を記述するシーンデータベースを保持する。なお、本実施形態では、ネットワーク102はイーサネット（登録商標）上に構築したLANである。

【0014】

ここで、以下の説明で用いる「データベース操作」という用語について説明しておく。データベースの「操作」とは、共有データベースを持つ任意のプロセスにおけるデータベースの内容を書き換える処理のことを差す。データベース操作は「操作指令」と「操作実行」の2段階に分かれる。操作指令はデータベースの更新要求であり、実際の書き換えは行われない。データベースに対する実際の書き換えは、操作実行の段階で行われる。以下の説明では特に断らない限り、単に「操作」と記述すれば「データベース操作」のことを指すものとする。操作対象がデータベース以外の場合は、「ユーザによる対話デバイスの操作」などのように操作対象を明示するものとする。

【0015】

なお、操作指令はユーザが発生させることもあれば、プロセスが動作の過程で発生することもある。前者の例としては、ユーザがマウスなどの対話デバイスを操作して、仮想オブジェクトを移動させた際に操作指令が発生されることが挙げられる。一方、後者の例としては、シューティングゲーム・システムにおいて、ゲームプログラムが敵キャラクタをアルゴリズム的に移動・回転させた際に操作指令が発生されることが挙げられる。

【0016】

さて、図2はデータベースを更新するための基本的な情報の流れを示すものである。図2で、A, B, C, D (201~204) はクライアント、X (205) はサーバである。いま、クライアントA (201) がシーンデータベースの操作指令を行ったとする。操作指令の内容はネットワーク経由でサーバに送信される（図2で(1)の矢印の過程）。サーバは、この操作指令を受けてクライ

アント A (201) から送信されたデータを、すべてのクライアントに配信する (図 2 で (2) の矢印の過程)。このような過程を経ることによって、非操作者のクライアント B (202)、C (203)、D (204) も操作の内容を知ることができる。

【0017】

次に、クライアントまたはサーバプロセスが動作する端末の構成を図 3 に示す。図 3 において、301 は CPU であり、端末全体の動作を制御する。302 はメモリであり、CPU 301 の動作に用いるプログラムやデータを格納する。303 はバスであり、各構成モジュール間のデータ転送を司る。304 はバス 303 と各種装置とのインタフェースである。また、305 はネットワークと接続するための通信部、306 は CPU 301 に読み込むプログラムやデータを格納する外部記憶装置、307 および 308 はプログラムを起動したり、プログラムの動作を指定するための入力装置を構成するキーボードおよびマウス、309 はプロセスの動作結果を表示するための表示部、310 は端末外部との入出力部である。例えば、図 27 に示すスタイラス装置が接続される。

【0018】

本実施形態のシステムは、データベースの更新タイミングに関して下記の 3 つのモードを持つ。以下、これらのモードを総称して「更新モード」と呼ぶ。

- (1) ウェイトモード (待機モード)
- (2) イミディエートモード (即時モード)
- (3) デイレイモード (遅延モード)

以下、各モードの処理について詳しく説明する。

【0019】

「ウェイトモード」は、ある操作指令に対する操作実行が、サーバからのイベントの配信を待って行われるモードである。図 4 は、ウェイトモードにおける時間の進行に沿った処理手順を模式的に示したものである。図 4 で、「操作者」は操作指令を行ったクライアント、「非操作者」は操作者以外のクライアントを指す。処理の進行は以下の説明の通りである。

【0020】

まず、手順T401で操作者クライアントがデータベースの操作指令を発する。この時点では、操作者クライアントのデータベースは更新されない。次に手順T402にて、T401の操作内容を示すイベントが生成され、T403でサーバに送信される。サーバは手順T404でイベントを受信し、T405で当該サーバとの接続が確立しているすべてのクライアントに向けてイベントを送信する（接続の確立について図11のステップS1102で後述する）。サーバより発行されたイベントを操作者クライアントが手順T406で受信し、その内容を分析して（T407）、データベースを書き換える（T408）。イベントの受信からの手順は、非操作者クライアントでも同様である（手順T409～T411）。図4に示した通り、ウェイトモードでは、操作指令（T401）から操作実行（T408）までタイムラグがある。しかし、ネットワーク上の情報転送の条件が同様であれば、操作者・非操作者クライアントともほぼ同時刻にデータベースが更新され、仮想空間の画像がクライアント間で同期することが期待できる。

【0021】

「イミディエートモード」は、操作者の指令が直ちにデータベースに反映され、これに対して非操作者のデータベースは遅れて更新されるものである。図5は、イミディエートモードにおける時間の進行に沿った処理手順を模式的に示したものである。処理の進行は以下の説明の通りである。

【0022】

まず、手順T501で操作者クライアントがデータベースの操作指令を発する。次に、手順T502で操作指令の内容に従ってデータベースが更新される。その後、操作内容を示すイベントが生成され（T503）、サーバに送信される（T504）。イベント送信後、操作者クライアントでは手順T501の操作指令および実行が行われたことを示す情報を操作キューとして保持しておく（T505、「操作キュー」については後述する）。

【0023】

サーバは手順T506でイベントを受信し、T507ですべてのクライアントに向けて当該イベントを送信する。操作者クライアントは手順T508でサーバからイベントを受信すると、操作キューを参照する（T509）。ここで、手順

T505で記録された情報により、すでに当該操作が実行済みであることが認識されるので、操作実行処理は行わず、手順T505で記録された情報を削除する(T514)。

【0024】

一方、非操作者クライアントでは手順T510で受信したイベントが未実行であることを操作キューを参照することにより認識し(T511)、イベントの内容に基づいて操作を実行する(T512, T513)。

【0025】

上記のような手順でデータベースを更新すると、操作者クライアントにおいては仮想空間に対する操作の結果がCG画像に反映されるまでの時間遅れがウェイトモードに比べて少なくなる。したがって、イミディエートモードは主に、ユーザが対話デバイスを用いて仮想物体を操作する場合に適している。なぜなら、一般的にユーザの対話的操作とその結果の提示の間に時間的ずれが生ずると、操作がより困難になる、ユーザに不快感を与えるなどの不具合が生じるからである。

【0026】

「ディレイモード」は、ウェイトモードのように操作指令を直ちに実行しないものの、操作実行の期限を設けて、その期限が過ぎた場合にはサーバからのイベント返信を待たずに当該操作を実行するというものである。図6は、ディレイモードにおいて、操作実行期限に到達する前に、サーバからのイベントが到着した場合の時間の進行に沿った処理手順を模式的に示したものである。処理の進行は以下の説明の通りである。

【0027】

まず、手順T601で操作者クライアントがデータベースの操作指令を発する。この時点では、データベースは更新されない。次に手順T602にて、T601の操作の内容を示すイベントが生成され、T603でサーバに送信される。イベント送信後、操作者クライアントでは手順T601の操作指令が行われたことを示す情報を操作キューに保持し(T604、「操作キュー」については後述する)、さらにタイマー処理を起動する(T605)。タイマー処理はタイムアウトを待って操作実行を行なうものであり、詳細は図19により後述する。

【0028】

一方、サーバは手順T606でイベントを受信し、T607ですべてのクライアントに向けてイベントを送信する。操作者クライアントは手順T608でイベントを受信し、操作キューを参照する(T609)。ここで、操作が未実行であることが認識されるので、当該操作キューに操作が実行済みであることを記録し(T610)、イベントの内容を分析し(T611)、データベースを書き換える(T612)。さらに、T604で記録された情報を削除する(T617)。また、非操作者クライアントでは手順T613で受信したイベントが未実行であることを認識し(T614)、イベントの内容に基づいて操作を実行する(T615, T616)。

【0029】

一方、図7はディレイモードにおいて、サーバからのイベントが到着する前に操作実行期限(タイムアウト)に至った場合の時間の進行に沿った処理手順を模式的に示したものである。処理の進行は以下の説明の通りである。

【0030】

まず、手順T701で操作者クライアントがデータベースの操作指令を発する。この時点では、データベースは更新されない。次に手順T702にて、T701の操作の内容を示すイベントが生成され、T703でサーバに送信される。イベント送信後、操作者クライアントでは手順T701の操作指令が行われたことを示す情報を操作キューに保持し(T704、「操作キュー」については後述する)、さらにタイマー処理を起動する(T705)。なお、タイマー処理はタイムアウトを待って操作実行を行なうものであり、詳細は図19により後述する。一方、サーバは手順T706でイベントを受信し、T709ですべてのクライアントに向けてイベントを送信する。

【0031】

タイマー処理では、サーバから対応するイベントを受信するまでの間にタイムアウト時刻に至ると、それをタイマー処理が検知し、手順T707で操作を実行して、操作キューに当該操作が実行済みであることを記憶する(T708)。したがって、操作者クライアントは、後に手順T710でイベントを受信するもの

の、操作キューを参照することにより当該操作が実行済みであることを認識し（T711）、操作履歴を更新する（T712（実行済みかつサーバプロセスから配信済みのイベントに対応する操作キューアイテムを削除））。非操作者クライアントでは手順T713で受信したイベントが未実行であることを認識し（T714）、イベントの内容に基づいて操作を実行する（T715、T716）。

【0032】

上記のような手順でデータベースを更新すると、イベント配信のタイムラグが大きければ操作実行が極力遅れないように動作し、タイムラグが小さければクライアント間の同期を図るように動作する。このような動作の切り換えは自動的に行われ、タイムアウト時間を適宜設定することにより、どちらの動作を優勢にするか調整することができる。

【0033】

図8は、クライアントからサーバに対して、或いはサーバからクライアントに対して発行されるイベントの内容を示したものである。イベントのデータは複数のフィールドから構成される。クライアントID801は、各クライアントを一意に識別するための番号で、クライアントがサーバとネットワーク接続を確立する際にサーバより割り当てられたものである（図11のステップS1102により後述する）。操作ID802は操作を一意に識別するための番号で、各クライアントに操作指令を行った時点で割り当てられる。このクライアントIDおよび操作IDの組を用いることにより、システム内部で起こったすべての操作指令を一意に特定することが可能である。

【0034】

更新モードID803は、各更新モードに対して一意に決定された識別番号で、このイベントに対応する操作指令の更新モードにより決定する。更新モードはデータベースの各エントリに設定、登録されている（図24により後述する）ので、これを参照して更新モードが設定される。エントリID804は、データベースの各エントリを一意に識別するための番号で、データをファイルからロードした際に割り当てられる。操作内容805は、操作指令の具体的内容であり、例えばCGオブジェクトのX座標をある値に設定する操作であれば、X座標属性を

識別する番号と X座標の設定値の組で表される。

【0035】

次に、操作キューについて説明する。操作キューは、イベントの受信または実行処理を待機している操作指令のリストである。ある時点で待機状態にある操作指令がm個あるとすると、操作キューは図9のように操作指令に関するm個の情報（キューアイテム）を指令の発生順に並べたものとなる。

【0036】

1つのキューアイテムの内容は図10のように、3つのデータフィールドから構成される。操作ID1001は、操作を一意に識別するための番号で、対応するイベントの操作IDフィールドと同じ値である。操作ID1001は、クライアントにおいて操作指令を行った時点で割り当てられる。タイムアウト時刻1002は、デイレイモードにおける操作実行期限であり、クライアントにおいて操作指令を行った時点で、現在時刻に所定のタイムアウト時間を加算して求められる。実行済みフラグ1003は、イミディエートモードあるいはデイレイモードにおいて、この操作IDを持つ操作が実行済みであるかどうかを示す。実行済みフラグは、操作指令を行った時点ではOFFに設定され、操作が実行された時点でONに変更される。

【0037】

なお、図10ではフィールド数3の操作キューアイテムを示したが、実行済みフラグをとタイムアウト時刻を兼用することも可能である。この場合、タイムアウト時刻が特殊な値、例えばマイナス1、である場合に実行済みであると定義すればよい。

【0038】

次に、クライアント、サーバのそれぞれについて、その処理の流れを詳しく説明する。

【0039】

図11は、クライアントの処理の全体的な流れである。クライアントが起動されると、ステップS1101にて仮想空間を記述するファイルを外部記憶装置306（図3）より読み込み、メモリ302（図3）上にシーングラフデータベ-

スを構築する。クライアント間でのデータの同一性を維持するため、ここでロードするファイルはクライアント間で同じ内容を持つとする。次に、サーバとのネットワーク接続を確立する（ステップS1102）。このとき、サーバから各クライアントを識別するためのクライアントIDが付与される。なお、クライアントとサーバ間のデータ交換は、TCP/IPプロトコルを用いた1対1のソケット通信にて行う。したがって、サーバは少なくともクライアントの数だけのソケット通信路を持つことになる。

【0040】

続いて、ステップS1103でプロセスを分岐し、操作指令を処理するための操作処理（ステップS1104）およびサーバから受信するイベントを処理するための受信イベント処理（ステップS1106）を起動する。なお、図11には図示していないが、シーングラフデータベースを参照して、仮想空間のCG画像を生成するプロセスも起動する。この処理は、公知のCG画像生成方法と同様であるのでその詳細な説明は省略する。また、操作処理および受信イベント処理については、後に詳しく説明する。操作処理および受信イベント処理はそれぞれステップS1105およびステップS1107でプロセスの終了する指示があったかどうか判定され、終了指示がなければステップS1104およびステップS1106に戻って操作指令および受信イベントを継続して処理する。一方、終了指示があればステップS1108でプロセスを統合し、サーバとのネットワーク接続を切断（ステップS1109）した上で、データベースの内容を外部記憶装置（図3の306）にファイルとしてセーブ（ステップS1110）して、すべての処理を終了する。

【0041】

ここで、ステップS1104の操作処理について図12を参照して詳しく説明する。初めに、ステップS1201にて操作指令の内容を入力する。このとき、操作指令を一意に特定する操作ID番号を決定する。次に、ステップS1202およびステップS1203で、データベースのエントリを参照して更新モードを判定する。更新モードに応じて、ステップS1206のウェイトモード操作処理、ステップS1205のイミディエートモード操作処理、ステップS1204の

ディレイモード操作処理、のいずれかのステップを経た後、処理を終了する。以下、更新モードごとの操作処理について、図13～図15を参照して詳述する。

【0042】

図13に、ウェイトモード操作処理の流れを示す。ステップS1301にて、ステップS1201で入力した操作の内容に基づいて、イベントデータ（図8参照）を生成する。そして、ステップS1302に進み、生成したイベントデータをサーバに向けて発信する。

【0043】

イミディエートモード操作処理の流れは図14に示す通りである。まず、ステップS1401にて、ステップS1201で入力した操作指令を実行する。具体的には、データベースの指定エントリの設定値を変更する。次に、ウェイトモードと同様にイベントを生成し（ステップS1402）、これをサーバに向けて発信（ステップS1403）する。最後のステップS1404で、操作IDをステップS1201で決定した操作ID番号に、実行済みフラグをONに、それぞれ設定した操作キューアイテムを操作キューに追加する。このとき、タイムアウト時刻は処理に用いないので、任意に設定してよい（例えば0）。

【0044】

次に、ディレイモード操作処理について図15を用いて説明する。まずウェイトモードと同様にイベントを生成し（ステップS1501）、これをサーバに向けて発信（ステップS1502）する。次に、操作IDをステップS1201で決定した操作ID番号に、タイムアウト時刻を現在時刻に所定のタイムアウト時間を加えた値に、実行済みフラグをOFFに、それぞれ設定した操作キューアイテムを操作キューに追加する（ステップS1503）。最後に、操作ID、操作内容、タイムアウト時刻のデータを渡してタイマープロセスを起動する（ステップS1504）。タイマープロセスについては、後に詳述する。

【0045】

続いて、ステップS1106（図11）の受信イベント処理について詳しく説明する。図16は、受信イベント処理の流れを示した図である。受信イベントは、不図示のイベント通信部がサーバより受信し、受信イベントバッファに入力さ

れる。

【0046】

ステップS1601では、受信イベントバッファを検索する。続くステップS1602で、受信イベントバッファにイベントが入力されているかどうか判定し、受信イベントがあればステップS1603に進む。一方、受信イベントが存在しないと判定されれば、処理を終了する。受信イベントがある場合、そのイベントデータを解析し、操作指令の内容を抽出する（ステップS1603）。次に、ステップS1604およびステップS1605でイベントの更新モードIDフィールドを参照して、操作指令の更新モードを判定する。そして、判定した更新モードに応じてステップS1608（ウェイトモードの場合）、ステップS1606（イミディエートモードの場合）、ステップS1607（ディレイモードの場合）のいずれかのステップを経た後、処理を終了する。ステップS1608では、イベントに記述された操作を実行する。具体的には、指定されたIDのエントリに関して、操作内容の通り設定値を変更する。以下、ステップS1606のイミディエートモードイベント処理、およびステップS1607のディレイモードイベント処理について、それぞれ図17、図18を参照して詳しく説明する。

【0047】

イミディエートモードイベント処理の流れを図17に示す。まず、ステップS1701にて操作キューを検索し、ステップS1702で操作キューに受信イベントに対応する操作キューアイテムが存在するかどうか判定する。具体的には、受信イベントのクライアントIDと本処理が動作しているクライアントのIDが一致し、かつ受信イベントの操作IDと操作キューアイテムの操作IDが一致すれば、その操作キューアイテムの操作指令と受信イベントの操作指令は対応していると判定する。イミディエートモードでは、対応するキューアイテムが存在する場合、その操作が当該クライアントで生成された時点で実行済みである。したがって、ここでは操作を実行せずに、ステップS1703で当該操作キューエントリを削除し、処理を終了する。一方、ステップS1702で対応するキューアイテムがないと判断された場合、その操作は当該クライアント以外のクライアントで生成されたものなので、まだ実行されていない。したがって、ステップS1

704で当該操作を実行し、処理を終了する。

【0048】

ディレイモードイベント処理の流れは図18に示す通りである。まず、ステップS1801にて操作キューを検索し、ステップS1802で受信イベントに対応する操作キューアイテムが操作キューに存在するかどうか判定する。ステップS1802の処理の詳細は、ステップS1702と同様である。ここで、対応する操作キューアイテムが存在する場合は、ステップS1803で操作キューアイテムの実行済みフラグをチェックする。実行済みフラグがONである場合は、タイムアウト時刻が過ぎているので、タイマープロセスが操作を実行済みである。したがって、ここでは操作を実行せず、ステップS1805にて当該操作キューエントリを削除し、処理を終了する。一方、ステップS1803にて、実行済みフラグがOFFと判定された場合は、未だタイムアウト時刻に到達していないため、操作は実行されていない。そこで、ステップS1806で当該操作を実行し、操作キューより当該アイテムを削除した（ステップS1805）後、処理を終了する。また、ステップS1802にて対応する操作キューアイテムが存在しないと判定された場合、その操作は当該クライアント以外のクライアントで生成されたものなので、まだ実行されていない。そこで、ステップS1804で当該操作を実行し、処理を終了する。なお、ステップS1804と、ステップS1806は、同じ内容の処理である。

【0049】

次に、図15のステップS1504で起動したタイマープロセスについて図19を用いて詳しく説明する。これは、ディレイモードにおいてタイムアウト時刻に到達した場合に対処するための処理である。

【0050】

まず、ステップS1901で操作キューを検索し、ステップS1902でタイマープロセス起動時に指定された操作に対応する操作キューアイテムが存在するかどうか判定する。具体的には、本プロセス起動時の引数である操作IDと操作キューアイテムの操作IDが一致すれば、その操作キューアイテムの操作指令と引数で指定された操作指令は対応していると判定する。対応するキューアイテム

が存在する場合、その操作指令に対応する受信イベントが未処理、すなわち実行されていないことを意味する。この場合は、タイムアウト時刻に到達しているかどうか確認する処理に進む。すなわち、ステップS1903で現在時刻を参照し、タイムアウト時刻との前後関係を判定する（ステップS1904）。もし、タイムアウト時刻を過ぎていれば、ステップS1905で操作を実行し、対応する操作キューアイテムの実行済みフラグをONに設定する。一方、タイムアウト時刻に達していない場合は、ステップS1901に戻る。また、ステップS1902にて、対応する操作キューアイテムが存在しないと判定された場合は、その操作指令に対応する受信イベントが処理済み、すなわち当該操作は実行済みであることを意味する。この場合は、操作の実行や操作キューの変更は必要ないため、そのまま処理を終了する。

【0051】

以上、クライアント側プロセスの流れを説明した。続いて、サーバ側プロセスの流れを図20を用いて詳しく説明する。

【0052】

最初にステップS2001で、クライアントからの接続要求を受け付け、通信を確立する。このとき、通信を確立した各クライアントにクライアントIDを通知する。次に、受信イベントバッファを検索し（ステップS2002）、受信イベントが存在するかどうか判定する（ステップS2003）。受信イベントがあれば、ステップS2004に進み、なければステップS2005に進む。ステップS2004では、接続している各クライアントに当該イベントを送信する。ステップS2005では、ユーザからの指令（例えば、キーボードのいずれかのキーを終了ボタンとして割り当て、そのキーが押下されたことをもってユーザからの終了指令とする）などによりサーバ処理を終了するかどうかを判定し、終了する場合はステップS2006に進み、終了しない場合はステップS2002に戻る。ステップS2006では、接続している各クライアントに処理を終了することを通知し、次にクライアントとの接続を終了して（ステップS2007）から処理を終了する。

【0053】

次に、本実施形態でクライアントが読み込むデータベース記述ファイルの構成について説明する。

【0054】

図24はエントリ数がNのデータのファイルフォーマットを模式的に示したものである。各エントリの属性は、スタートデリミタおよびエンドデリミタで囲まれたフィールドに記述される。図24で、1番目、2番目、N番目のエントリのスタートデリミタはそれぞれ2401、2406、2409で示されている。一方、それぞれのエンドデリミタは2405、2408、2411である。エントリの属性は、2402～2404のように、各属性を識別するための識別子および属性値の組で記述される。更新モードおよびタイムアウト時間は、上記のフォーマットのファイルでエントリごとに規定されている。ただし、該当エントリの更新モードがディレイモードでない場合は、タイムアウト時間は記述しなくてもよい。また、いずれの更新モードにおいても両属性（更新モードおよびタイムアウト時間）を必ずしも記述する必要はなく、もし記述がなければ、メモリ（図3の302）上にシーングラフデータベースを構築する時点（ステップS1101）で、予めシステムによって決められたデフォルト値に設定される。

【0055】

なお、上述のデータベース記述ファイルでは、選択する更新モードを記述している。これに対して、選択不可能な更新モードを1つまたは複数設定することもできる。このような選択不能な更新モードの設定の活用については、第5実施形態による更新モードの変更操作において詳述する。

【0056】

また、上記の説明では、サーバ・クライアント間のデータ通信を1対1のTCP/IPソケットにて行うとしているが、ブロードキャストあるいはマルチキャストによりイベントを交換してもよいし、通信プロトコルはTCP/IPに限らない。

【0057】

また、通信媒体としてはイーサネット（登録商標）を利用しているが、USBやFirewireなど他の情報伝達媒体でも構わない。更に、ネットワークの形態もL

ANだけでなく、WAN経由の接続でもよいし、LANとWANを併用しても構わない。

【0058】

また、サーバの数は、1つの共有データベースシステムにつき1つに限定する必要はなく、複数のサーバが存在してもよい。この場合、各サーバに処理を割り当てる形態は任意である。例えば、クライアントごとに異なるサーバを用いてもよいし、あるいはデータベースのエントリごとにサーバを設けてもよい。

【0059】

更に、端末に対するプロセスの割当は1端末1プロセスに限定されるものではなく、1つの端末上で複数のプロセスを動作させることができる。その場合、プロセスの種別の組み合わせは任意であり、クライアントのみ、サーバのみ、クライアントおよびサーバの、いずれの組み合わせでもシステムは動作する。なお、同じ端末上のプロセスは、ネットワークに替えて、共有メモリを経由してデータを交換してもよい。

【0060】

また、イベントはサーバが仲介して各端末に配信されるが、データ操作を行った端末が直接に他の端末にイベントを送信してもよい。なお、サーバを経由しない場合、例えば図4のT406において、操作クライアントは自分自身からイベントを受け取ることになる。これは、イベントの送信宛て先を自ホストに設定することにより実現できる。なお、自ホストと他ホストを区別せずに通信する機能は主要なOSには標準的に備わっているものである。

【0061】

また、上記の説明では、3次元仮想空間データを共有するとしているが、データベースの内容はこれに限らず任意であることはいうまでもない。また、データベース全体を共有する場合のみならず、その一部のみを共有する場合にも上記実施形態で説明した方法が適用可能であることはいうまでもない。これらの特徴は、以下の実施形態についても同様である。

【0062】

以上のように第1実施形態によれば、情報伝達媒体を介して接続された複数の

プロセス（クライアントプロセス）によって共有される共有データ（データベース）を該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法が開示される。特に、共有データに対する操作要求が発生した場合には、該操作要求を表す要求情報（イベント）が情報伝達媒体上に出力され、この要求情報に対する応答情報（イベント）を情報伝達媒体より受信し、受信した応答情報に従って共有データに対する操作を実行する。このようなウェイトモードによれば、共有データに対する操作が情報伝達媒体を介して受信されるイベントに従って実行されるので、共有データの更新タイミングを高精度に一致させることができる。

【0063】

また、ディレイモードでは、タイマー処理により、応答情報を受信する前に操作要求の発生から所定時間が経過した場合には、該応答情報の受信を待たずに当該操作要求に基づく共有データの操作が実行される。このため、操作が発生したプロセスにおける応答性と複数プロセス間の同期タイミングの一致の両立を図ることができる。

【0064】

また、本実施形態によれば、操作要求に応じて操作キューにキューアイテムを登録し、操作実行工程によって共有データに対する操作が実行された場合に対応するキューアイテムを処理済とするキュー制御を行い、操作キューにおいて応答情報が表す操作要求に対応するアイテムが処理済となっていない場合に、応答情報に対応する操作を共有データに対して実行する。上述のようなディレイモードにおいて、操作実行が重複することを簡易な構成で防止できる。

【0065】

また、上記実施形態によれば、受信工程によって応答情報が受信されるのを待って共有データの操作を実行する第1モード（ウェイトモード）と、操作要求の発生からの所定時間の経過と、応答情報の受信の何れかの早い方のタイミングで共有データの操作を実行する第2モード（ディレイモード）を含む複数の変更モードのいずれかで動作することが可能である。

【0066】

そして、共有データは複数のアイテムで構成され（図24）、複数のアイテムの各々は採用すべき変更モードを指定する指定情報を含む。変更モードはこの指定情報に従って選択され、共有データへの操作が実行されることになる。すなわち、複数のアイテム毎に変更モードが切り換えられることになり、この構成によれば、アイテム毎に適切な変更モードを設定できる。

【0067】

また、サーバプロセスにおいては、情報伝達媒体を介して接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するために、複数のクライアントプロセスとの接続を確立し、これら複数のクライアントプロセスより共有データの変更に関するイベントを受信する。そして、受信したイベントを、上記複数のクライアントプロセスに対して発行する。

【0068】

〔第2実施形態〕

第1実施形態では、すべての更新モードにおいて、サーバがすべてのイベントをすべてのクライアントに配信している。しかしながら、イミディエートモードの場合には、操作者クライアントは操作を実行済みであり、サーバからイベントを受信する必要はない。本実施形態では、このような場合にサーバからのイベントの配信を省略するものである。

【0069】

本実施形態におけるイベント配信処理の流れを図21に示す。この処理は、図20のステップS2004に替えて実行されるべきものである。

【0070】

まず、ステップS2101にて、更新モードがイミディエートモードかどうか判定する。もし、イミディエートモードであれば、イベントのクライアントIDフィールドを参照して、そのID以外のクライアントに対してイベントを送信する（ステップS2102）。一方、更新モードがウェイトモードあるいはディレイモードの場合は、第1実施形態と同様にすべてのクライアントに対してイベントを送信する（ステップS2103）。イベント発行元のクライアントプロセス

には、サーバプロセスからイベントが配信されないため、操作キューの参照および更新処理が不要となる。

【0071】

以上のように第2実施形態によれば、サーバプロセスによって受信されるイベントが共有データの変更手順を示す変更モード情報を含み、サーバはこの変更モード情報に基づいて当該イベントの配信先を制御する。例えば、受信したイベントの変更モード情報に基づいて当該イベントの発行元において共有データの処理が既に完了しているか判定し、完了していると判定された場合には、当該イベントをその発行元のクライアントへは発行しないように制御する。図21に示すように、変更モードがイミディエートモードの場合は、イベントを受信した時点で発行元のクライアントプロセスにおける操作実行は完了しているので、当該発行元クライアントプロセスへはイベントの配信を行なわない。

【0072】

上記の処理を行うことにより、サーバからクライアントに送信するイベントの総量が減るので、ネットワーク負荷を軽減することができる。また、イベント発行元のクライアントプロセスは、操作キューの処理が不要となるので、処理負荷を軽減することができる。

【0073】

〔第3実施形態〕

第1実施形態および第2実施形態では、更新モードがディレイモードの場合に、サーバが受信したイベントはタイムアウト時刻に関係なくクライアントに送信されている。しかしながら、タイムアウト時刻を過ぎると、操作者クライアントはイベントが返信されなくとも操作を実行するので、サーバからイベントを受信する必要はない。第3実施形態は、このような場合にイベントの配信を省略するものである。

【0074】

第3実施形態の場合には、図8のイベントの内容に替えて、図22の内容のイベントを用いる。これは、図8のイベントにタイムアウト時刻が追加されたものである。このタイムアウト時刻は、操作指令が生成された時刻にタイムアウト時

間を加えたものであり、操作者クライアントにおいてタイマー処理が操作実行を開始する判定基準とする時刻に等しい。

【0075】

第3実施形態におけるイベント配信処理の流れを図23に示す。この処理は、図20のステップS2004に替えて実行されるべきものである。

【0076】

まず、ステップS2301にて、更新モードがディレイモードかどうか判定する。もし、ディレイモードであれば、現在時刻を参照し（ステップS2302）、現在時刻とイベントに記述されたタイムアウト時刻の前後関係を判定する（ステップS2304）。もし、現在時刻がタイムアウト時刻よりも後ならば、発信者クライアントでは操作の実行を開始していると見なされる。したがって、イベントのクライアントIDフィールドを参照して、そのID以外のクライアントに対してイベントを送信する（ステップS2305）。一方、更新モードがディレイモード以外の場合や、タイムアウト時刻に到達していない場合は、ステップS2306に進んで、すべてのクライアントにイベントを送信する。

【0077】

上記の処理を行うことにより、サーバからクライアントに送信するイベントの総数が減るので、ネットワーク負荷を軽減することができる。

【0078】

なお、イベントのタイムアウト時刻とサーバの現在時刻を比較して、正確な前後関係を求めるには、クライアントとサーバの現在時刻が一致していなければならない。このための仕組みとしては、例えばNTP（Network Time Protocol）などが利用できる。

【0079】

以上のように、第3実施形態は、変更モードがディレイモードであった場合に、イベントの発行元クライアントプロセスで操作実行が完了しているかを判断可能とするしくみを提供する。すなわち、イベントにタイムアウト時刻を含ませ、変更モード情報がタイムアウト時刻の経過によって操作が実行される変更モード（ディレイモード）であることを示し、且つ、現在時刻がタイムアウト時刻を過

ぎている場合に、当該イベントの発行元において共有データの処理が既に完了していると判定する。

【0080】

以上のように、第2および第3実施形態では、端末間で交換されるデータベース更新情報が一部省略されるので、ネットワークへの負荷が軽減される。

【0081】

〔第4実施形態〕

第2実施形態はイミディエートモード、第3実施形態はディレイモードにおけるイベント配信処理であるが、これらを併用することも可能である。

【0082】

すなわち、第4実施形態では、第2および第3実施形態におけるネットワーク負荷の軽減方法を併用するものである。

【0083】

〔第5実施形態〕

上述の各実施形態では、更新モードがデータベース記述ファイルで設定されており、システムの稼働中に更新モードを変更することは考慮されていない。第5実施形態では、データベースの更新モードを変更する手段を有するデータベース更新方法について説明する。

【0084】

図25は、本実施形態におけるクライアントの画面表示例を示したものである。第1実施形態と同様に、データベースの内容は3次元仮想空間であり、表示ウィンドウ2501中の3次元空間表示部2502に、3次元仮想物体2503および2504のCG画像が表示されている。3次元表示部の下部には、仮想物体の位置、姿勢および更新モードを設定するための、位置設定部2506、姿勢設定部2507および更新モード設定部2508が設けられている。更新モード設定部2508には、各更新モードに対応したラジオボタンが表示されている。

【0085】

更新モードを設定する際には、まずカーソル2505をモードを変更したい仮想物体像の上に移動してマウス308（図3）の左ボタンをクリックすることに

より、その物体を選択する。このとき、その物体に設定されている更新モードに対応したラジオボタンが選択表示される。次に、更新モード設定部にあるラジオボタンのうち、所望のモードに対応するものをマウスの左ボタンでクリックして有効にする。このように、ユーザの対話的操作によって、モードをユーザの希望に応じて設定することができる。なお、位置設定部 2506 および姿勢設定部 2507 には、スライダが表示されており、これ进行操作することによって選択した仮想物体の位置および姿勢を対話的に変更することができる。

【0086】

図 26 は、別の表示例及びユーザインターフェース例を示したものである。ここでは、ユーザが HMD (Head Mounted Display) を頭部に装着し、図 26 のような仮想空間の 3 次元表示画面 2601 を観察している。画面中で、2602 および 2603 は仮想物体、2604 は物体を選択・操作するためのポインタである。ユーザは図 27 に示すようなスタイラス装置 2701 を手に持ち、これを動かすことによりポインタ 2604 の位置や姿勢を制御する。なお、スタイラス装置は、指し棒のような形状を有し、位置姿勢センサとボタンを備えている。例えば、ボタンを押した時点での棒の先端部の位置・姿勢を計測するという目的で利用される。

【0087】

更新モードを変更する際には、まずポインタ 2604 がモードを変更したい仮想物体像を指すようにスタイラス装置 2701 を動かして、スタイラス装置 2701 上の選択ボタン 2702 を押下することにより、その物体を選択する。このとき、ユーザが観察する画面上には選択した物体の属性を表すアノテーション表示 2605 が現れる。アノテーション表示の中には、現在選択されている更新モードも記されている。次に、スタイラス装置の端部に設けられたダイヤル 2703 を回転させる。回転に従って、アノテーション表示の更新モードが入れ替わるので、所望のモードが表示された時点でダイヤル 2703 の回転を止める。このように、ユーザの対話的操作によって、モードをユーザの希望に応じて設定することができる。なお、HMD は表示部 309 (図 3) に相当するものである。また、スタイラス装置 2701 は入出力部 310 とデータ伝送ケーブル 2704 を

介して接続し、CPU301が選択ボタン2702の選択状態・ダイヤル2703の回転状態を読み出す。

【0088】

また、図25や図26のように仮想空間を3次元表示するのに替えて、2次元表示を用いてもよい。図28は2次元表示の例である。この例では、表示ウィンドウ2801の中の2次元表示部2802に、3次元空間をある方向から観察した様子が平面状に表示されている。2次元表示部の中には、仮想物体2803および2804がアイコンで表示され、カーソル2805をアイコン上に置いてマウス308の左ボタンをクリックすることにより、物体を選択できる。更新モードを設定する際には、物体を選択した後、所望の更新モードに対応してシステムで設定されているキーボード307の特定キーを押下する。このように、ユーザの対話的操作によって、更新モードをユーザの希望に応じて設定することができる。なお、物体を選択した際に、画面上に選択した物体の属性を表すアノテーション表示2806を表示してもよい。アノテーション表示の中に、現在選択されている更新モードも記せば、現在の更新モードを確認しながら更新モードの設定操作を行うことができる。

【0089】

以上、更新モードの設定方法を3例説明した。ただし、上述のいずれの更新モード設定方法においても、選択不可能な更新モードが指定されていれば、ユーザがそのモードを選択しようとしても、実際に変更されないように構成してもよい。

【0090】

なお、更新モードは共有データベースのエントリがもつ属性なので、あるクライアントが更新モードを変更した場合、その変更が他のクライアントにも反映されることになる。すなわち、更新モードの変更も操作指令の一つとして扱われ、イベントによって各端末に反映される。

【0091】

一方、更新モードの設定をクライアント間で共有せず、各クライアントごとに設定できるようにすることも可能である。その場合の操作処理の流れを図29に

示す。これは、図12の処理に替えて用いるものである。

【0092】

図29の処理では、ステップS2901にて操作指令を入力した後、ステップS2902において、入力された操作指令がモード設定であるかどうか判定する。モード設定であると判定された場合は、ステップS2903に進み、指定のエントリに関する更新モードの属性値を操作指令に従って設定する。その他のステップS2904、S2905、S2906、S2907およびS2908は、それぞれ図12のステップS1202、S1203、S1204、S1205およびS1206と同様である。

【0093】

以上のように第5実施形態によれば、共有データ中の所望のアイテム（図24のエントリ）に対応するオブジェクト表示を選択して所望の変更モードを指定するユーザインターフェースが提供される（図25等）。このユーザインターフェースにより、アイテム毎に更新モードを所望に設定できる。

【0094】

また、更新モードの設定操作によるモードの切り換えは、複数のクライアントの共有データに反映されるように構成してもよいし、当該クライアントの共有データのみ反映されるようにしてもよい。更に、更新モードの設定（切り換え結果）を共有する／しないを設定可能としてもよい。

【0095】

〔第6実施形態〕

第5実施形態では更新モードを変更する手段を説明したが、タイムアウト時間を変更する手段を追加することも可能である。第6実施形態では、タイムアウト時間が変更可能である実施形態を示す。

【0096】

例えば、図25の表示画面にタイムアウト時間を調整するスライダを追加したり、図27のスタイラス装置にタイムアウト時間を設定するダイヤルを装備することで実施できる。あるいは、キーボード307の所定のキーを押下すると、タイムアウト時間を加算、減算するように構成してもよい。

【0097】

いうまでもなく、更新モードの変更手段およびタイムアウト時間の変更手段のうち、どちらか片方を備えてもよいし、両方を備えてもよい。

【0098】

〔第7実施形態〕

上述の実施形態では、データベースのエントリごとに更新モードおよびタイムアウト時間を設定しているが、クライアントごとに更新モードやタイムアウト時間を設定可能としてもよい。また、データベースのエントリの属性ごとに更新モードやタイムアウト時間を設定できるようにすることも可能である。

【0099】

〔第8実施形態〕

データベースシステムによっては、データを更新する処理と、データを参照する処理が一定の順序で行われるものがある。このような場合には、ある操作指令から実行までに時間遅れがあるとしても、いずれかのデータ更新過程で実行が完了すれば、その直後の参照には操作の結果が反映される。

【0100】

一般的に、仮想空間の描画システムも上記のようなシステムの1つであり、シーングラフの更新と、シーンの描画が交互に起こる。図30は、これを概念的に示したもので、“Update”が更新過程、“Draw”が描画過程を表している。ここで、更新過程と描画過程の組をフレームと呼ぶ。

【0101】

ここで、操作指令に対する結果がデータベースに反映されるまでの遅れに関して、要求仕様が2フレーム未満であるとする。いま、S番目のフレームの時刻 t_1 に1つの操作指令が発生したとする。その操作が時刻 t_3 までに実行されれば、要求仕様を満たしているといえる。これを実現するためには、例えばタイムアウト時刻を t_2 として操作指令をデイレイモードで処理すればよい。しかしながら、一般的には時刻 t_2 を推定することは困難である。

【0102】

そこで、第8の実施形態では、操作指令を実行するタイミングを自動的に決定

することが可能な実施形態を示す。

【0103】

上述したタイマー処理（図19）でタイムアウト時刻を確認するためのステップS1903およびステップS1904に替えて、データベース更新過程の到来を検知する処理を加える。その処理で、更新過程の到来を検知したらステップS1905に進み、更新過程が到来していなければステップS1901に戻る。なお、更新過程の到来を検知するのではなく、図30の更新過程を開始した時点で、タイマープロセスに割り込みを掛けて、ステップS1905に移行させてもよい。

【0104】

以上のように第8実施形態によれば、応答情報を受信する前にシステムが定める更新過程の開始時刻に達した場合には、該応答情報の受信を待たずに前記操作要求に基づく前記共有データの操作が実行される。すなわち、タイムアウトのタイミングが自動的に設定され、所望の遅れ時間で操作指令を実行させることが可能になる。

【0105】

〔第9実施形態〕

上述の第5実施形態では、更新モードをすべてマニュアルで設定している。これに対し第9の実施形態では、更新モードを自動的に設定する実施形態を示す。本実施形態では、操作指令がユーザの対話的操作に基づくものである場合には、自動的にイミディエートモードあるいはディレイモードに設定する。

【0106】

これを実現するためには、対話的操作が行われた場合、操作指令を生成する処理の前に、更新モードをイミディエートモードあるいはディレイモードに設定する指令を加える処理を追加すればよい。このとき、イミディエートモードとディレイモードのどちらを設定するかは、対話的操作の種類ごとに予め決めておく。たとえば、図25のスライダや図27のスタイラスにより仮想物体を移動・回転する場合は、イミディエートモードに設定する。一方、マウスクリックや選択ボタン（図27の2702）の押下により物体を選択する場合は、ディレイモード

に設定する。

【0107】

以上のように、第9実施形態によれば、共有データのアイテムに対応するオブジェクトに対する操作内容に応じて変更モードが設定される。例えば、更新モードが自動的にイミディエートモードあるいはディレイモードに設定されるので、対話的操作の指令から操作実行までの時間遅れをウェイトモードに比べて少なくすることが可能であり、遅れに起因する不快感が軽減される。

【0108】

なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0109】

この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0110】

プログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0111】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0112】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0113】

【発明の効果】

以上説明したように、本発明によれば、複数のプロセスで共有される共有データに関して、操作を実行するタイミングを高精度に一致させることが可能となる。

【図面の簡単な説明】

【図1】

第1実施形態におけるデータベース共有システム全体の構成を示すブロック図である。

【図2】

第1実施形態によるプロセス間の基本的な情報伝達方法を示す図である。

【図3】

第1実施形態による端末装置のハードウェア構成を示すブロック図である。

【図4】

ウェイトモードにおけるデータベースの更新手順を示す図である。

【図5】

イミディエートモードにおけるデータベースの更新手順を示す図である。

【図6】

ディレイモード（非タイムアウト時）におけるデータベースの更新手順を示す図である。

【図7】

ディレイモード（タイムアウト時）におけるデータベースの更新手順を示す図である。

【図 8】

イベントデータのデータ構成例を示す図である。

【図 9】

操作キューのデータ構成例を示す図である。

【図 10】

操作キューアイテムのデータ構成例を示す図である。

【図 11】

第 1 実施形態によるクライアントプロセスのフローチャートである。

【図 12】

クライアントプロセスにおける操作処理を示すフローチャートである。

【図 13】

図 12 のウェイトモード操作処理を説明するフローチャートである。

【図 14】

図 12 のイミディエートモード操作処理を説明するフローチャートである。

【図 15】

図 12 のディレイモード操作処理を説明するフローチャートである。

【図 16】

図 11 のクライアントプロセスにおける受信イベント処理を示すフローチャートである。

【図 17】

図 16 のイミディエートモード受信イベント処理のフローチャートである。

【図 18】

図 16 のディレイモード受信イベント処理のフローチャートである。

【図 19】

タイマープロセスを説明するフローチャートである。

【図 20】

第 1 実施形態によるサーバプロセスを説明するフローチャートである。

【図 21】

第 2 実施形態におけるサーバのイベント配信処理を説明するフローチャートで

ある。

【図 2 2】

第 3 実施形態におけるイベントデータのデータ構成例を示す図である。

【図 2 3】

第 3 実施形態におけるサーバのイベント配信処理を説明するフローチャートである。

【図 2 4】

データベースの記述ファイルフォーマットを示す図である。

【図 2 5】

第 5 実施形態におけるクライアント表示画面の一例を示す図である。

【図 2 6】

第 5 実施形態におけるクライアント表示画面の一例を示す図である。

【図 2 7】

第 5 実施形態におけるスタイラス装置の外観と構成を示す図である。

【図 2 8】

第 5 実施形態におけるクライアント表示画面の一例を示す図である。

【図 2 9】

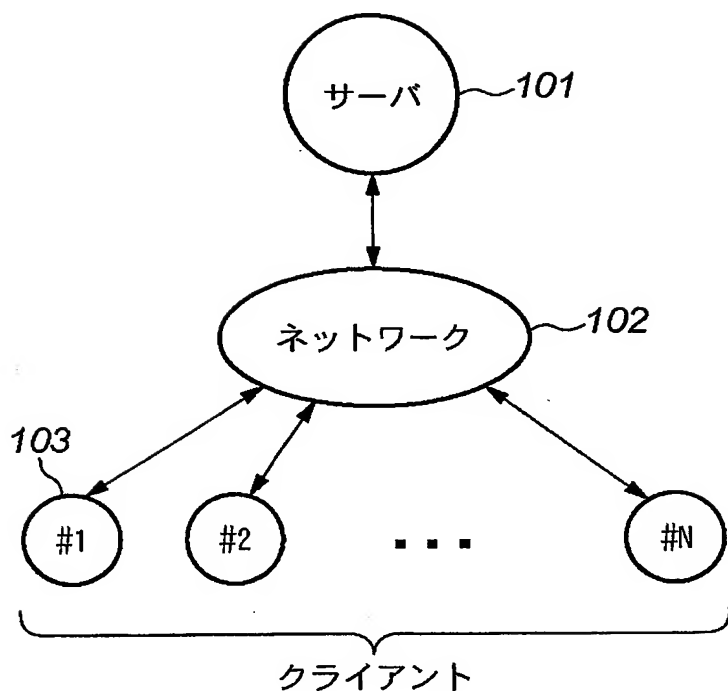
第 5 実施形態における操作処理を説明するフローチャートである。

【図 3 0】

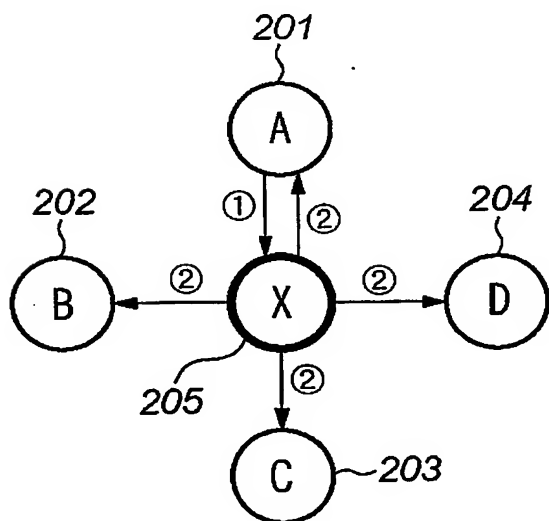
第 8 実施形態におけるタイムアウト時刻の設定方法を示す概念図である。

【書類名】 図面

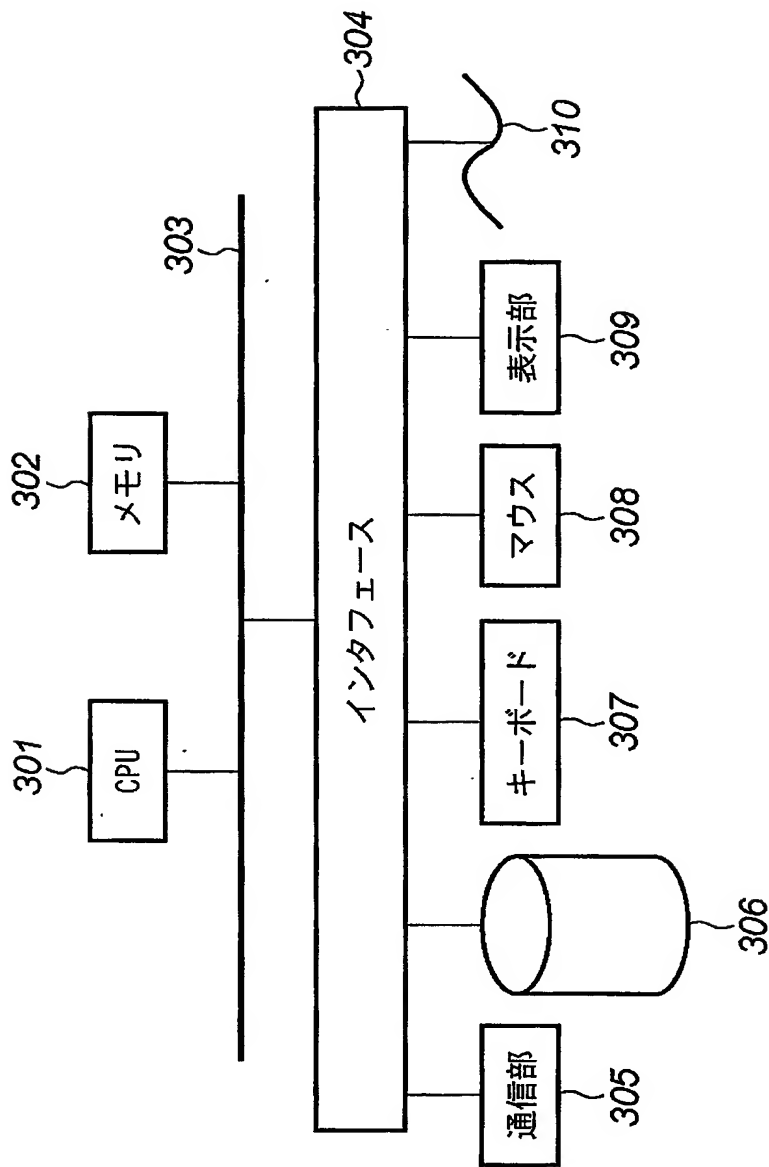
【図 1】



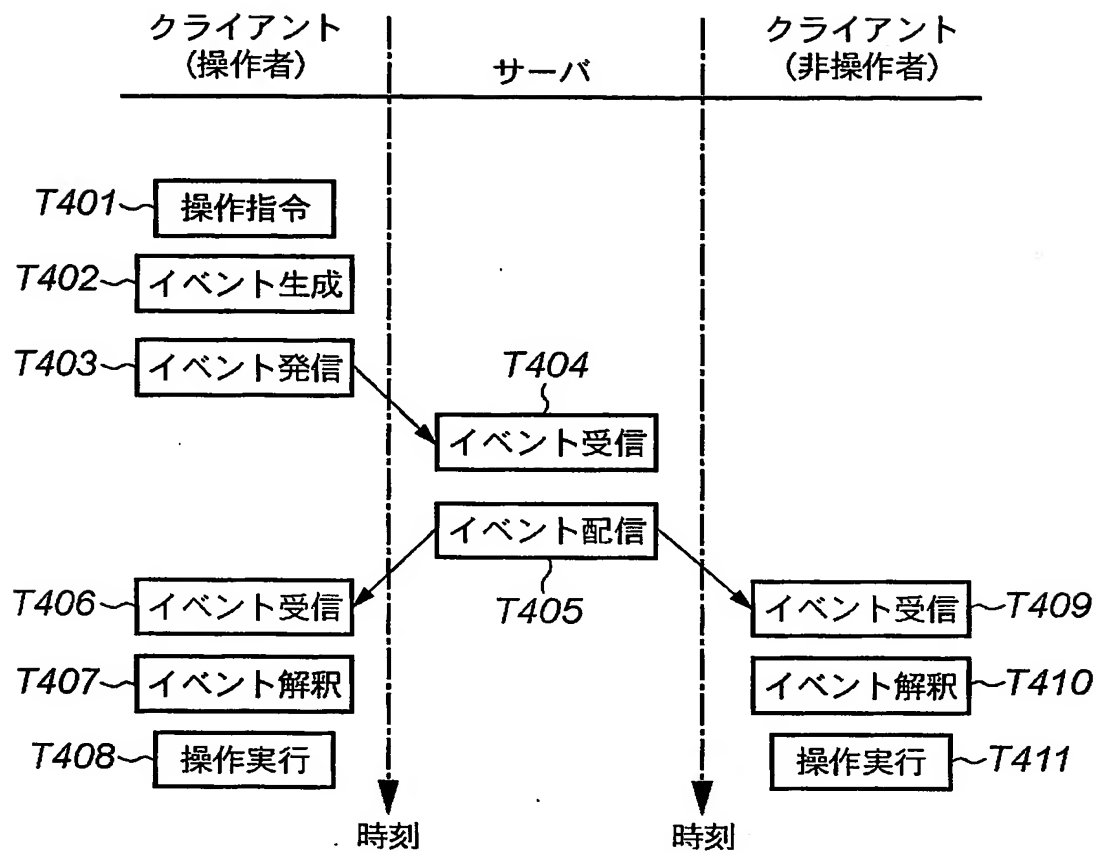
【図 2】



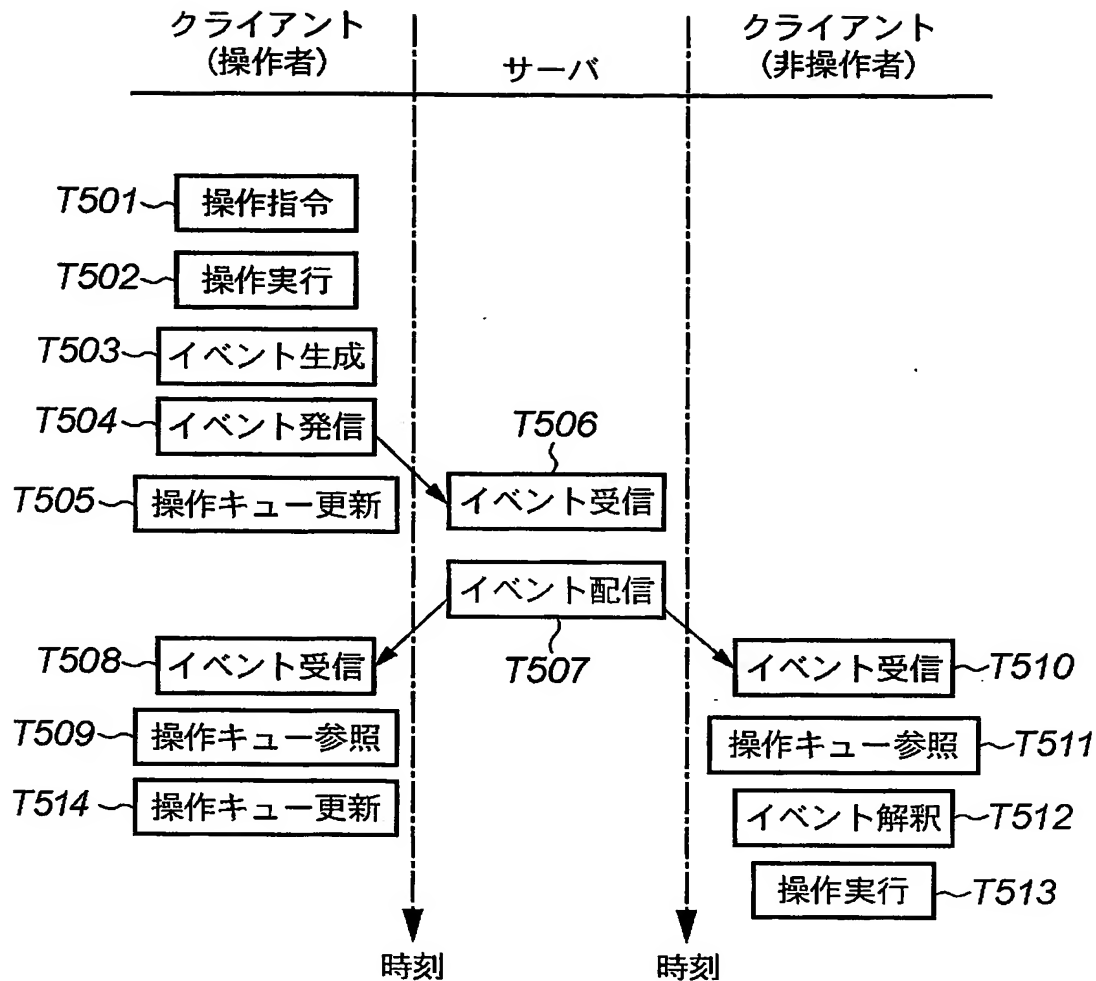
【図3】



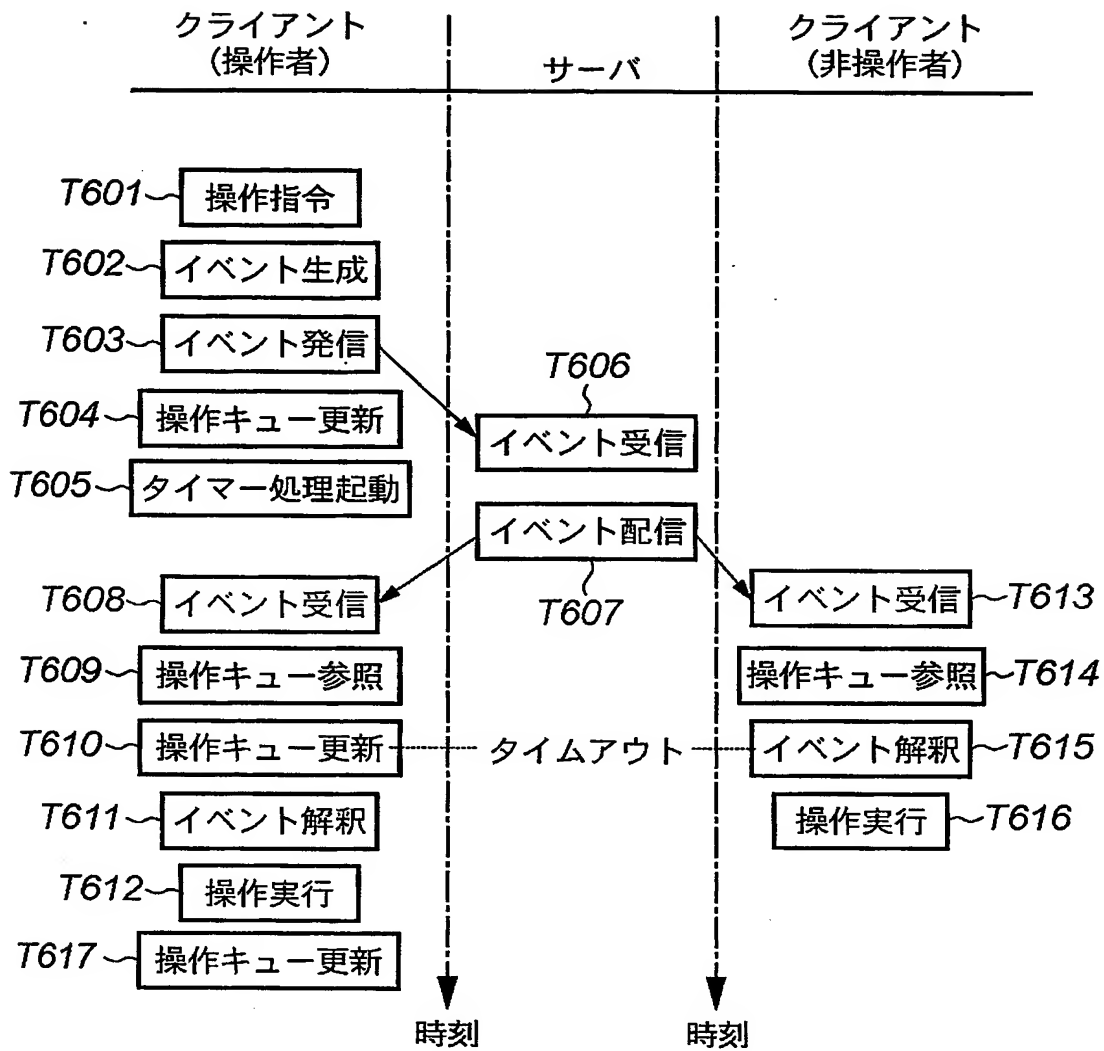
【図 4】



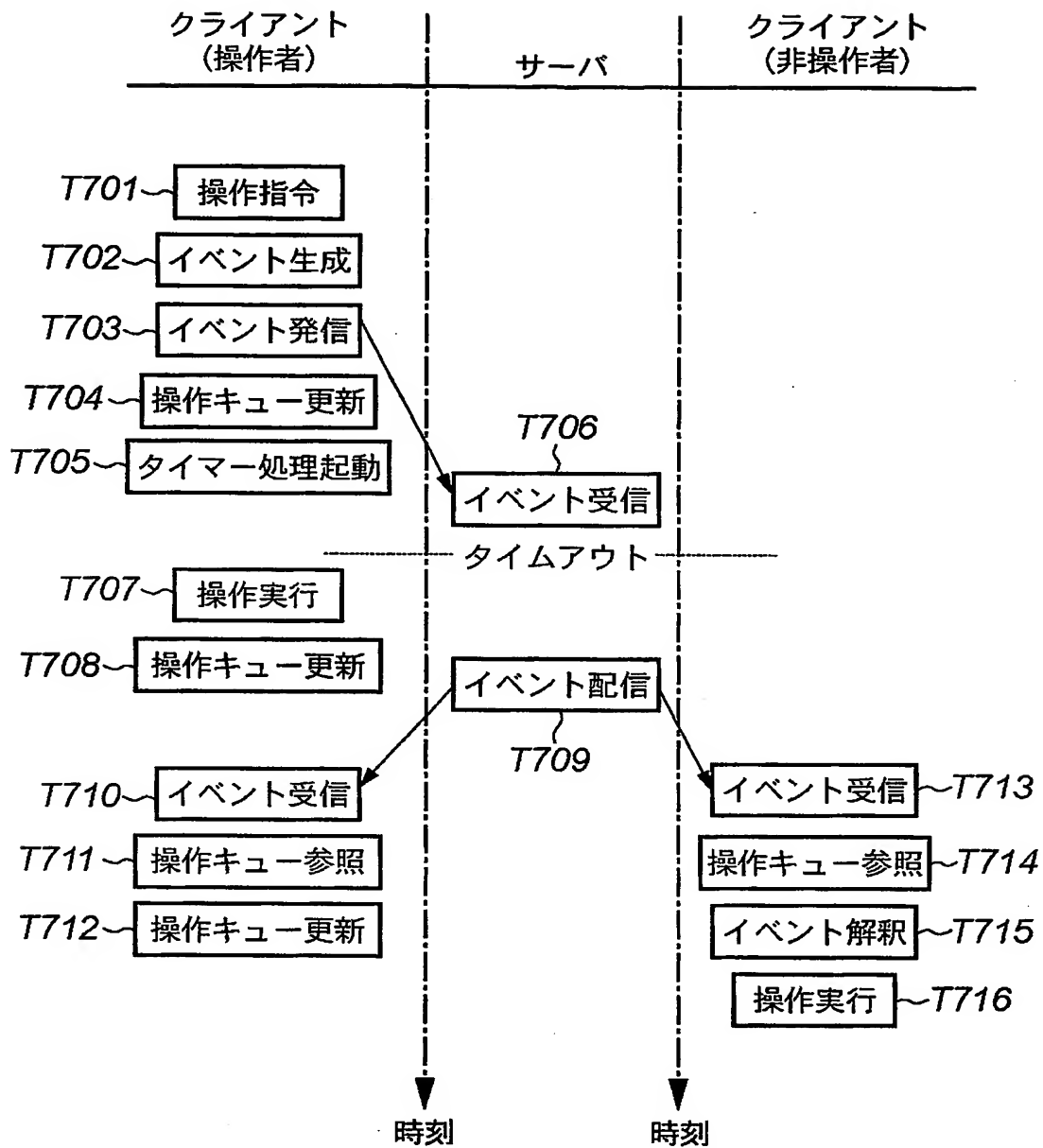
【図 5】



【図 6】



【図 7】



【図 8】

クライアントID	〜801
操作ID	〜802
更新モードID	〜803
エントリID	〜804
操作内容	〜805

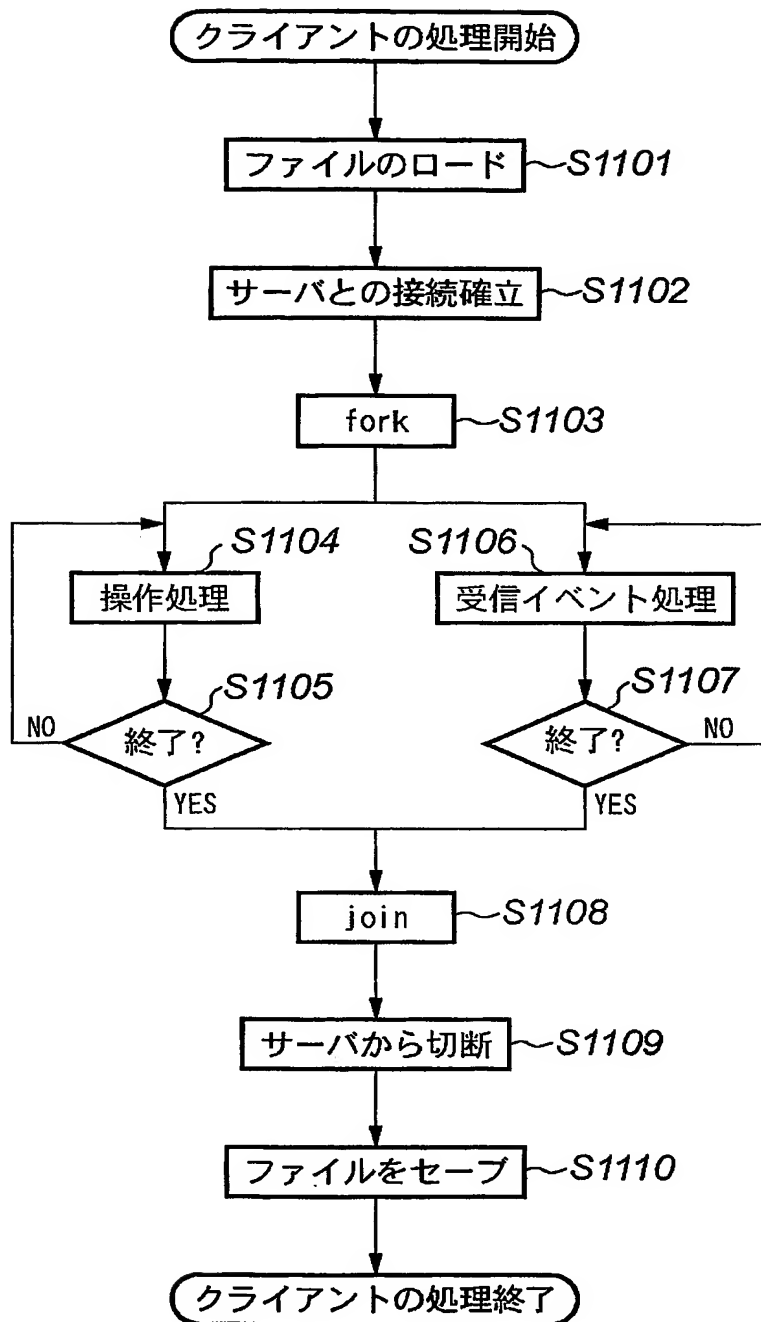
【図 9】

キューアイテム No. m
...
キューアイテム No. 2
キューアイテム No. 1

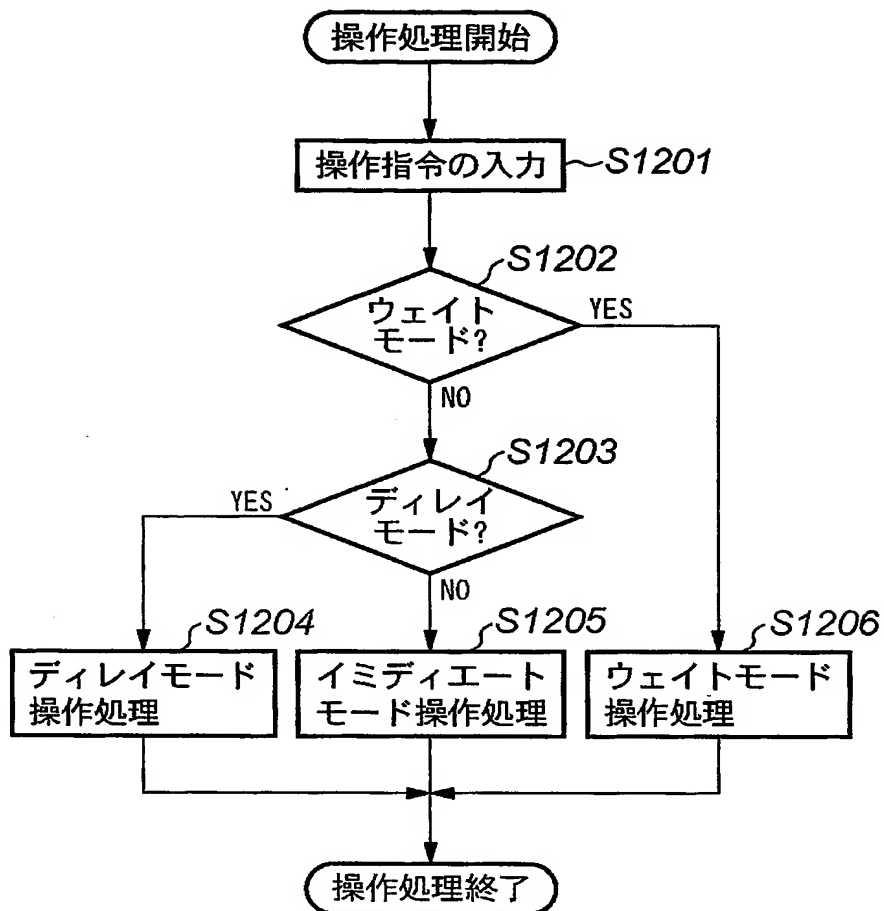
【図10】

操作ID	〜1001
タイムアウト時刻	〜1002
実行済みフラグ	〜1003

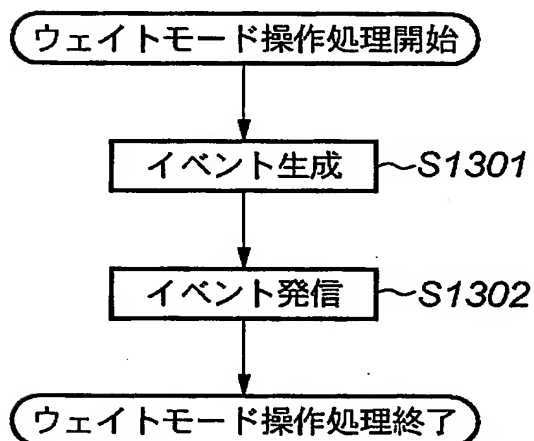
【図 11】



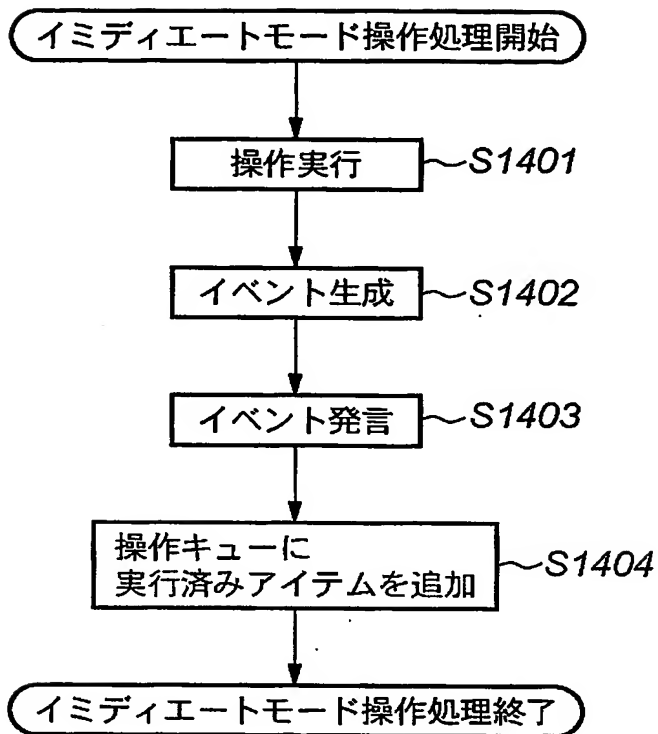
【図 12】



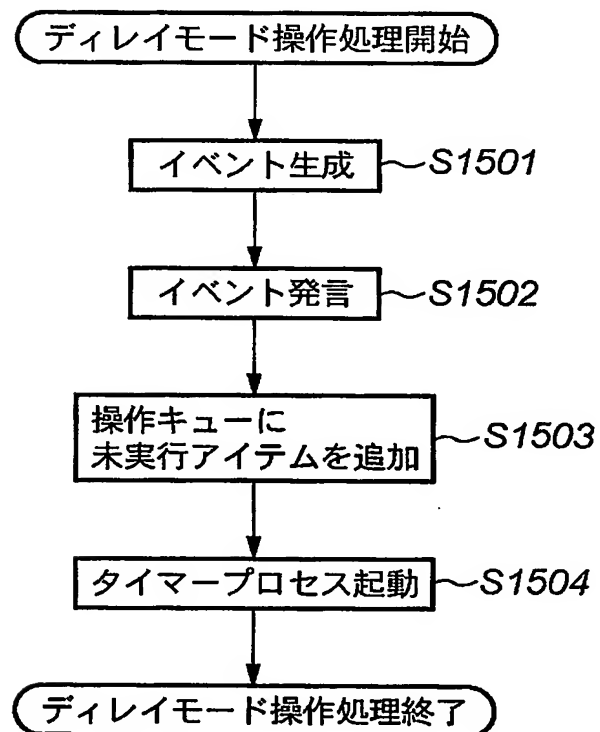
【図 13】



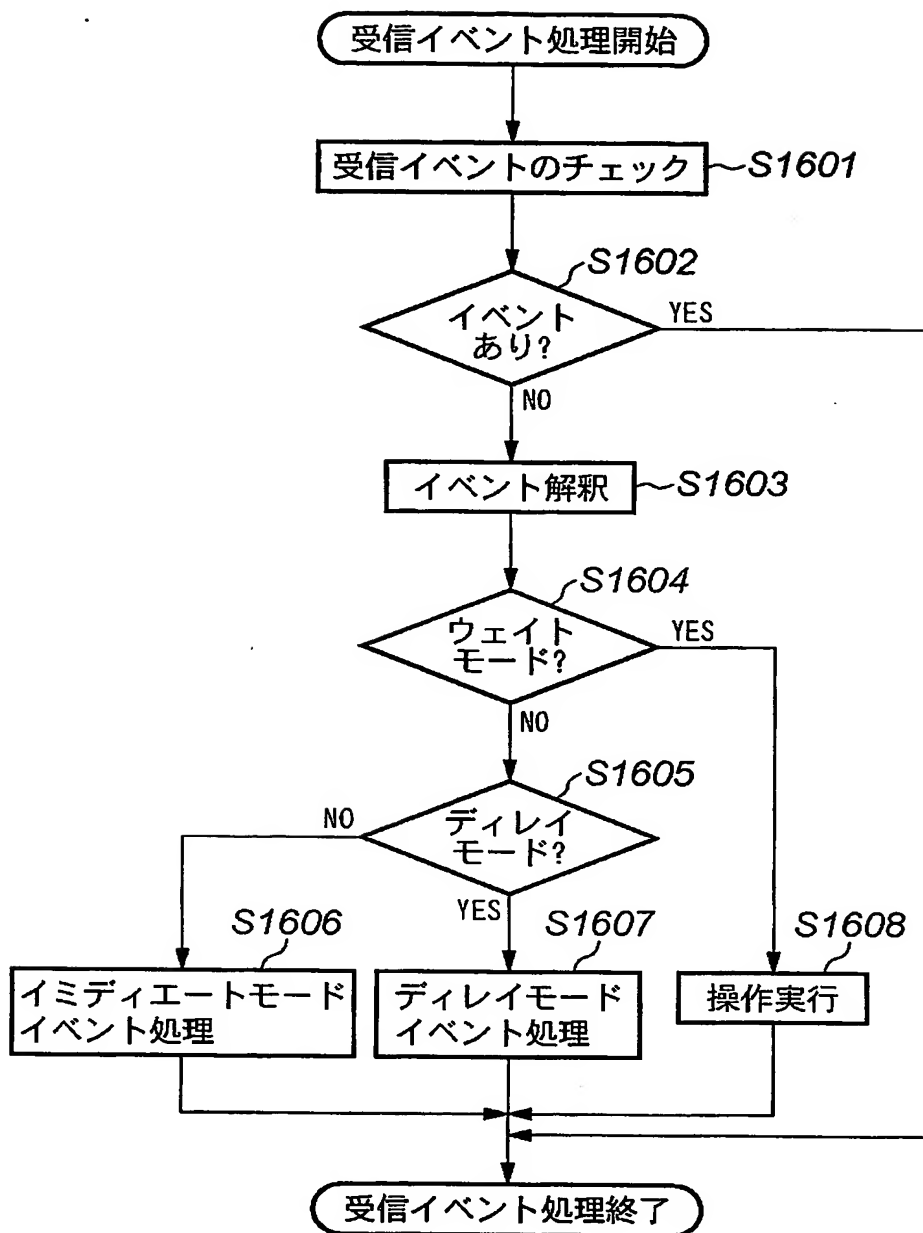
【図 14】



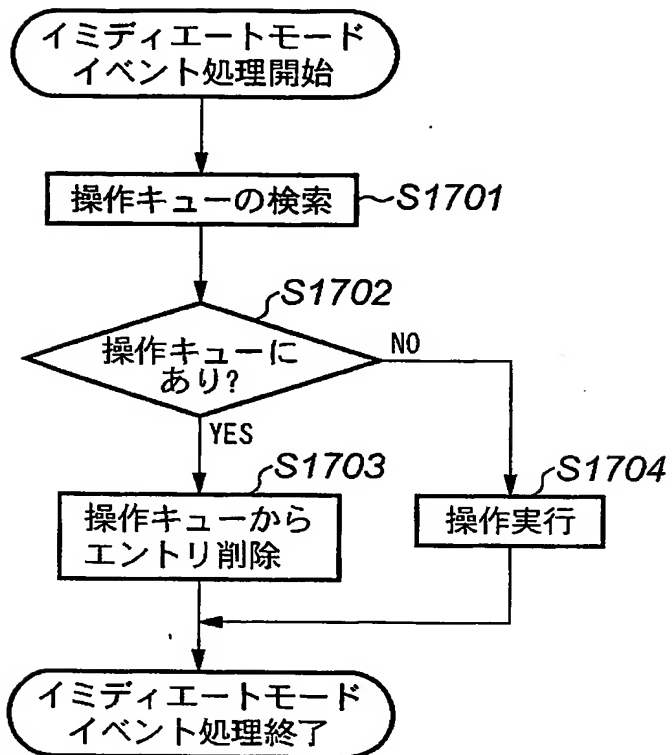
【図 15】



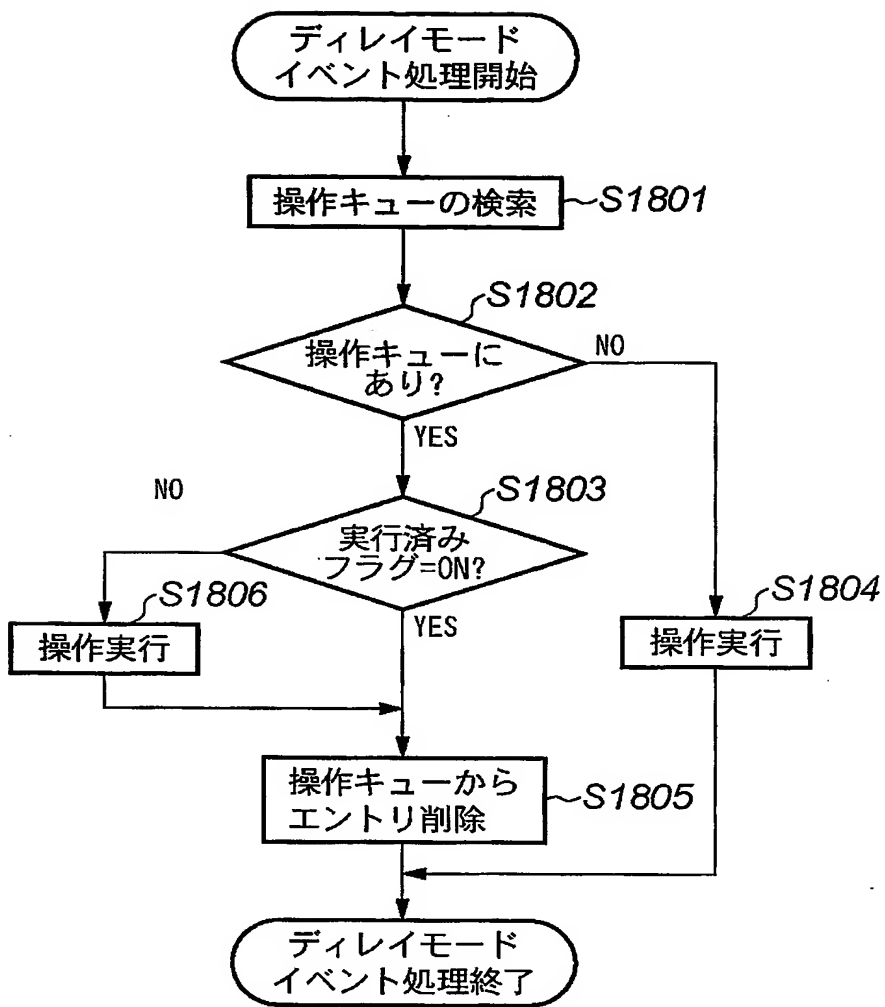
【図 16】



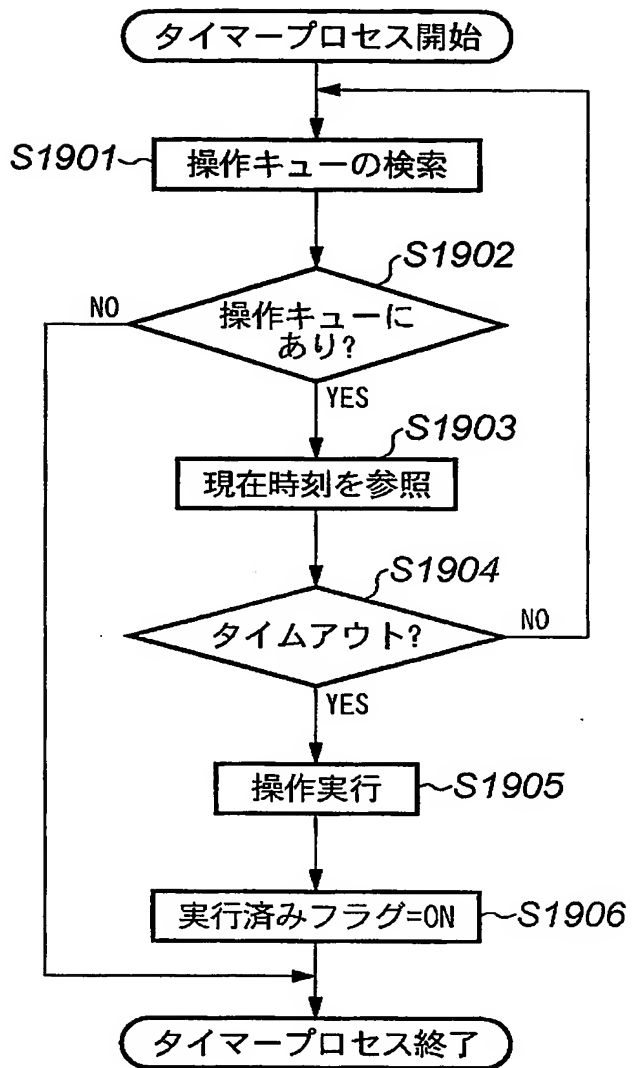
【図 17】



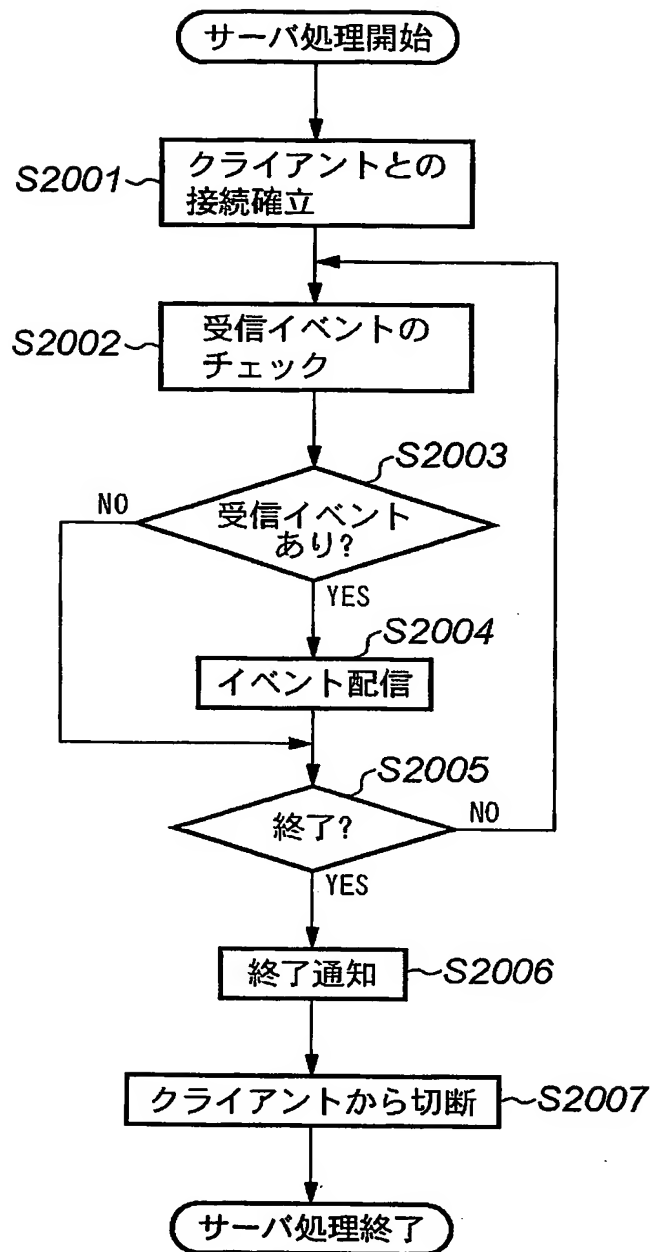
【図 18】



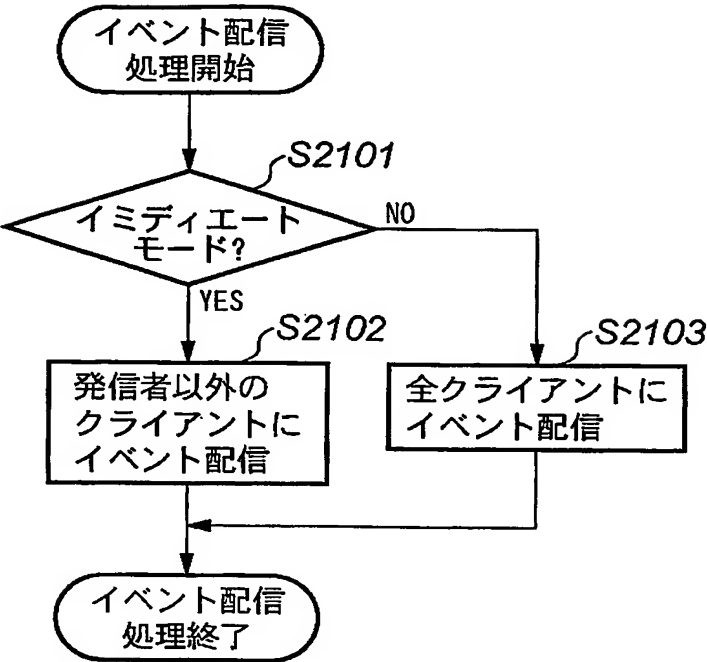
【図 19】



【図 20】



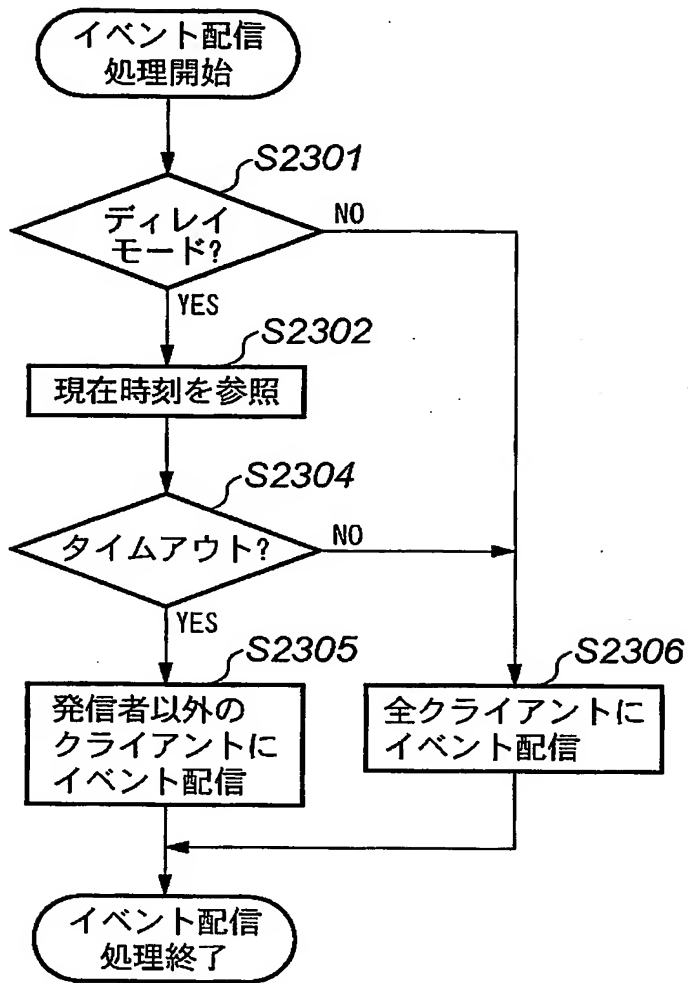
【図 2 1】



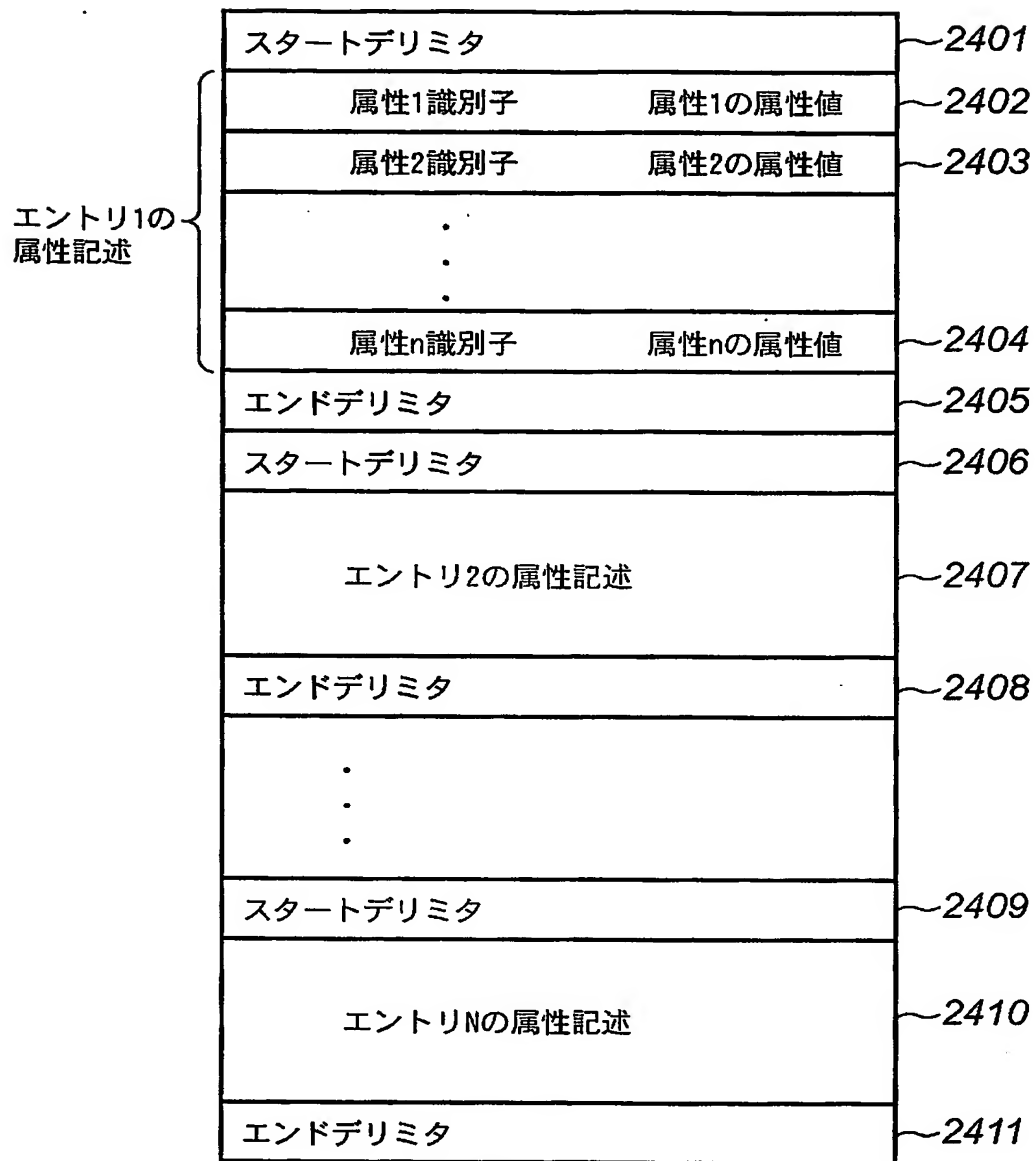
【図 2 2】

操作者 ID
操作 ID
更新モード ID
タイムアウト時刻
エントリ ID
操作内容

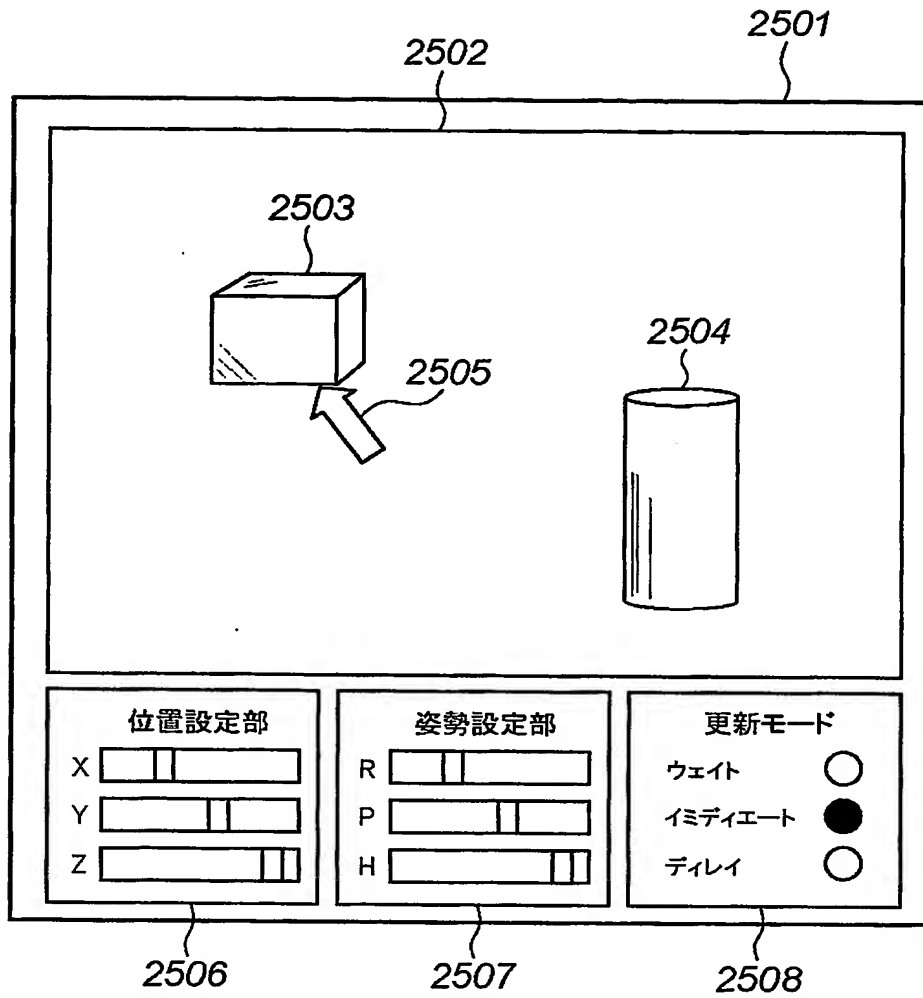
【図 23】



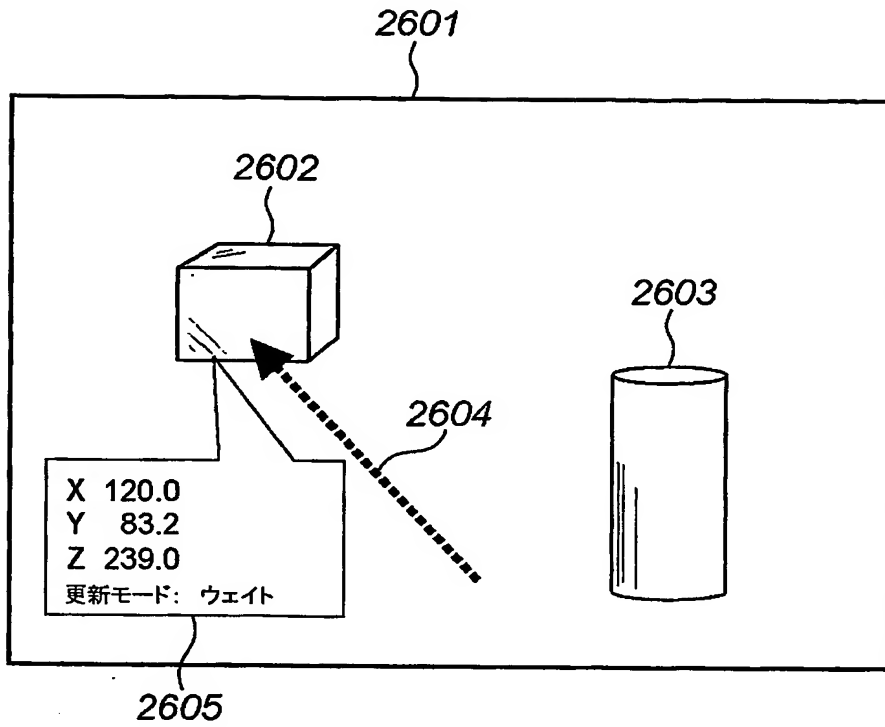
【図24】



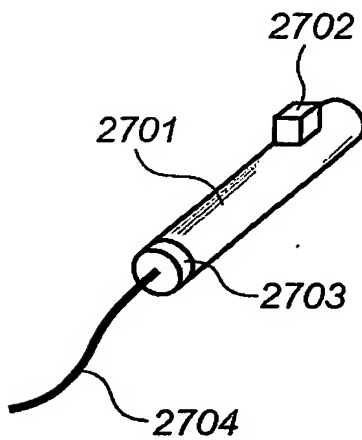
【図 25】



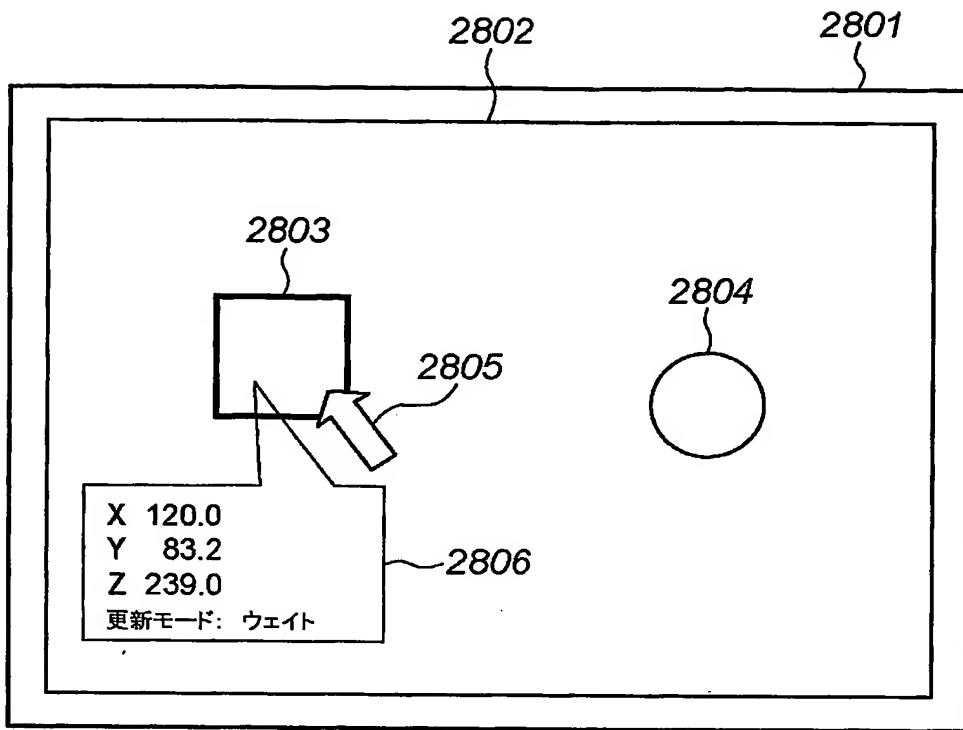
【図 26】



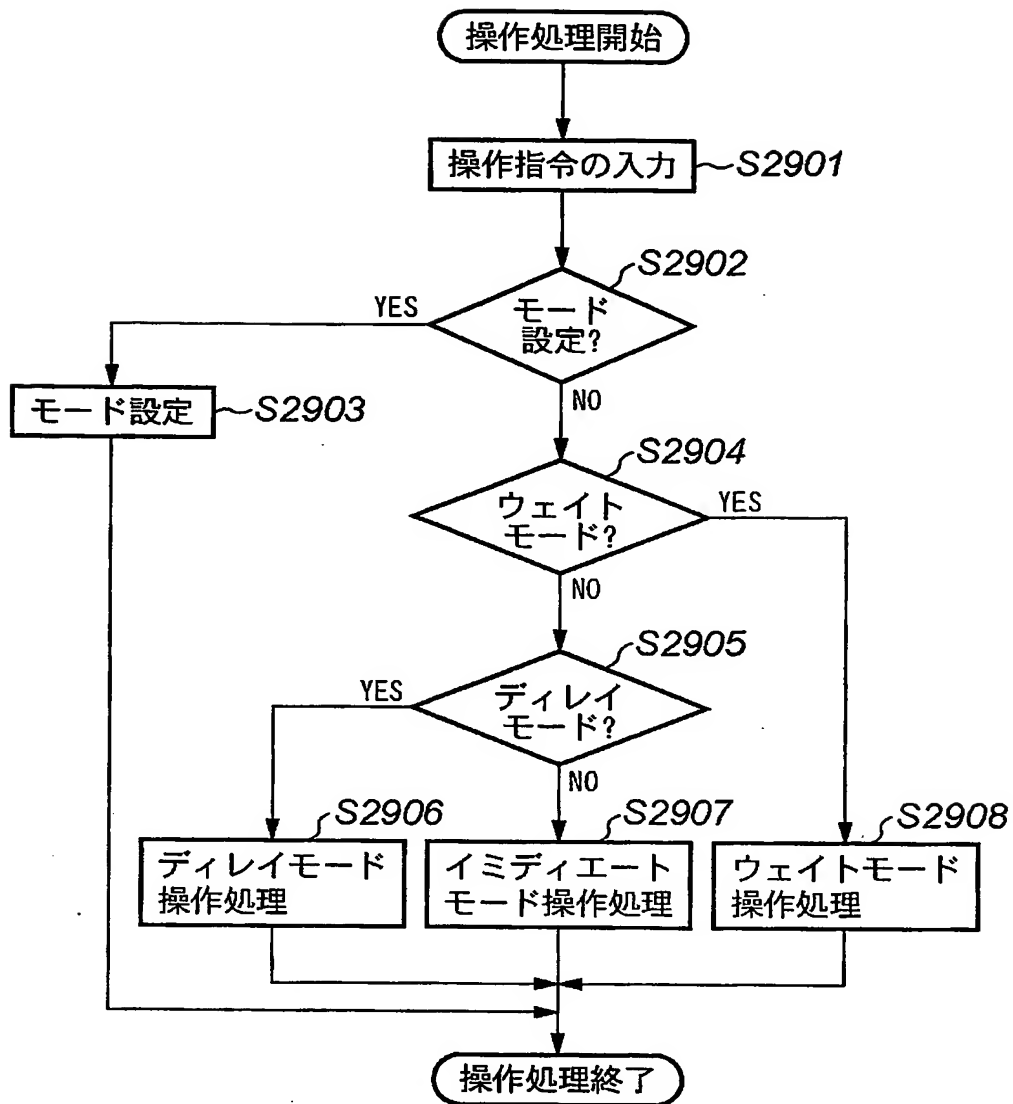
【図 27】



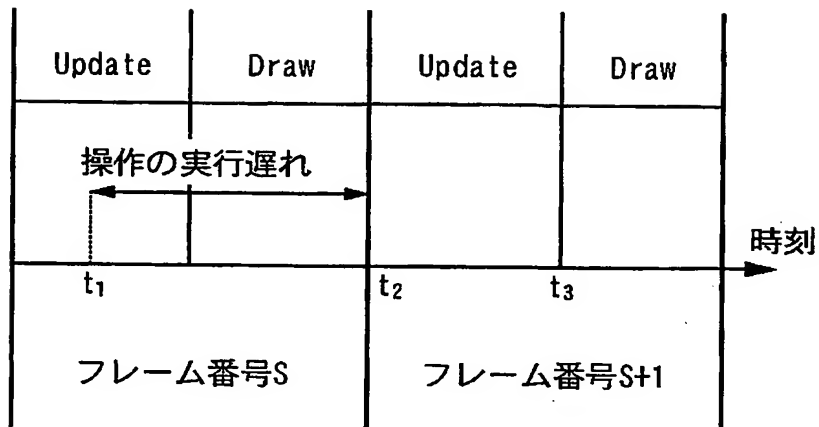
【図 28】



【図 29】



【図 30】



t_1 : 操作指令発生時刻, t_2 : タイムアウト時刻,
 t_3 : 操作の結果が反映される時刻

【書類名】 要約書

【要約】

【課題】 複数のプロセスで共有される共有データに関して、操作を実行するタイミングを高精度に一致させることを可能とする。

【解決手段】 ネットワークを介して接続された装置の複数のプロセス間で共有される共有データを該複数のプロセスの各々が保持するシステムにおいて、各プロセスが保持する共有データの同一性を維持するための情報処理方法において、共有データに対する操作要求があった場合に、該操作要求を表すイベントがネットワーク上に出力される。そして、ネットワーク上に出力されたイベントに対する返信のイベントをネットワークより受信する。そして、受信されたイベントに従って共有データに対する操作が実行される。

【選択図】 図 2

特願 2002-348726

出 願 人 履 歴 情 報

識別番号

[000001007]

1. 変更年月日

1990年 8月30日

[変更理由]

新規登録

住 所

東京都大田区下丸子3丁目30番2号

氏 名

キャノン株式会社